Term Project 390.4- 2019

## R Markdown

pacman::p\_load(dplyr, tidyr, ggplot2, magrittr, stringr, mlr)  
housing\_data = read.csv("housing\_data\_2016\_2017.csv")

##Delete variables that we dont need

housing\_data %<>%  
 select(-c(HITId, HITTypeId, Title, Description, Keywords, Reward, CreationTime, MaxAssignments, RequesterAnnotation, AssignmentDurationInSeconds, AutoApprovalDelayInSeconds, Expiration, NumberOfSimilarHITs, LifetimeInSeconds, AssignmentId, WorkerId, AssignmentStatus, AcceptTime, SubmitTime, AutoApprovalTime, ApprovalTime, RejectionTime, RequesterFeedback, WorkTimeInSeconds, LifetimeApprovalRate, Last30DaysApprovalRate, Last7DaysApprovalRate, URL, url, date\_of\_sale))

## Clean Data

housing\_data %<>%  
 mutate( zip\_code = str\_extract(full\_address\_or\_zip\_code, "[0-9]{5}"))   
  
housing\_data %<>%  
 mutate(dogs\_allowed = ifelse(substr(housing\_data$dogs\_allowed, 1, 3) == "yes", 1, 0)) %>%  
 mutate(cats\_allowed = ifelse(substr(housing\_data$cats\_allowed, 1, 3) == "yes", 1, 0)) %>%  
 mutate( pets\_allowed = ifelse( cats\_allowed + dogs\_allowed > 0, 1, 0)) %>%  
 mutate(coop\_condo = factor(tolower(coop\_condo)))  
  
housing\_data %<>%  
 select(-c(dogs\_allowed,cats\_allowed, fuel\_type))  
  
d = housing\_data  
  
d %<>%  
 mutate(maintenance\_cost = sjmisc::rec(maintenance\_cost, rec = "NA = 0 ; else = copy")) %<>%  
 mutate(common\_charges = sjmisc::rec(common\_charges, rec = "NA = 0 ; else = copy"))##recode from NA to 0.  
  
  
# combine maintaince cost and common charges  
d %<>%   
 mutate( monthly\_cost = common\_charges + maintenance\_cost)  
  
d %<>%  
 mutate(monthly\_cost = sjmisc::rec(monthly\_cost, rec = "0 = NA ; else = copy"))  
  
## Garage exists conver it to binary  
  
d %<>%  
 mutate(garage\_exists = sjmisc::rec(garage\_exists, rec = "NA = 0 ; else = copy")) ##recode from NA to 0.   
  
d %<>%  
 mutate(garage\_exists = sjmisc::rec(garage\_exists, rec = " eys = 1; UG = 1 ; Underground = 1; yes = 1 ; Yes = 1 ; else = copy")) ##recode from NA to 0.  
  
d %<>%  
 select(-c(maintenance\_cost , common\_charges, model\_type))  
  
  
str(d)

## 'data.frame': 2230 obs. of 24 variables:  
## $ approx\_year\_built : int 1955 1955 2004 2002 1949 1938 1950 1960 1960 2005 ...  
## $ 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 ...  
## $ dining\_room\_type : Factor w/ 5 levels "combo","dining area",..: 1 3 1 1 1 1 1 NA NA 5 ...  
## $ 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/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ kitchen\_type : Factor w/ 4 levels "combo","eat in",..: 2 2 3 2 2 2 3 3 2 2 ...  
## $ 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 ...  
## $ lat : num 40.7 40.8 40.7 40.8 40.7 ...  
## $ lon : num -73.8 -73.8 -73.9 -73.8 -73.7 ...  
## $ zip\_code : chr "11355" "11354" "11368" "11354" ...  
## $ pets\_allowed : num 0 0 0 0 1 1 0 0 0 0 ...  
## $ monthly\_cost : num 767 604 167 275 660 932 660 514 781 NA ...

##Change variable type

d %<>%  
 mutate( dining\_room\_type = as.factor(dining\_room\_type)) %>%  
 mutate(garage\_exists = as.character(garage\_exists)) %>%  
 mutate(garage\_exists = as.numeric(garage\_exists)) %>%  
 mutate( parking\_charges = as.character(parking\_charges)) %>%  
 mutate( parking\_charges = as.numeric(parking\_charges)) %>%  
 mutate(sale\_price = as.character(sale\_price)) %>%  
 mutate(sale\_price = as.numeric(sale\_price)) %>%  
 mutate(total\_taxes = as.character(total\_taxes)) %>%  
 mutate(total\_taxes = as.numeric(total\_taxes)) %>%  
 mutate(price\_persqft = listing\_price\_to\_nearest\_1000 / sq\_footage)

## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion

#Added latitude and longitude features using ggmap

#Already run and included in the data  
#pacman::p\_load(ggmap)  
#d %<>%  
# mutate(lat = geocode(full\_address\_or\_zip\_code)$lat, lon = #geocode(full\_address\_or\_zip\_code)$lon )  
#geocoordinates for relevant LIRR stations  
  
lirr\_coord = coord

## Error in eval(expr, envir, enclos): object 'coord' not found

RAD\_EARTH = 3958.8  
degrees\_to\_radians = function(angle\_degrees){  
 for(i in 1:length(angle\_degrees))  
 angle\_degrees[i] = angle\_degrees[i]\*pi/180  
 return(angle\_degrees)  
}  
compute\_globe\_distance = function(destination, origin){  
 destination\_rad = degrees\_to\_radians(destination)  
 origin\_rad = degrees\_to\_radians(origin)  
 delta\_lat = destination\_rad[1] - origin\_rad[1]  
 delta\_lon = destination\_rad[2] - origin\_rad[2]  
 h = (sin(delta\_lat/2))^2 + cos(origin\_rad[1]) \* cos(destination\_rad[1]) \* (sin(delta\_lon/2))^2  
 central\_angle = 2 \* asin(sqrt(h))  
 return(RAD\_EARTH \* central\_angle)  
}  
#find the closest LIRR station and compute distance  
shortest\_lirr\_distance = function(all\_lirr\_coords, house\_coords){  
 shortest\_dist = Inf  
 for (i in 1: nrow(all\_lirr\_coords)){  
 ith\_lirr = c(all\_lirr\_coords$lat[i], all\_lirr\_coords$lon[i])  
 new\_dist = compute\_globe\_distance(ith\_lirr, house\_coords)  
 if( new\_dist < shortest\_dist){  
 shortest\_dist = new\_dist  
 }  
 }  
 return(shortest\_dist)  
}  
d %<>%  
 rowwise() %>%  
 mutate(shortest\_dist = shortest\_lirr\_distance(lirr\_coord, c(lat, lon)) )

## Error in nrow(all\_lirr\_coords): object 'lirr\_coord' not found

#makes any other addresses redundant  
d %<>%  
 select(-c(zip\_code, full\_address\_or\_zip\_code, listing\_price\_to\_nearest\_1000))

We are trying to predict sale\_price. So let’s section our dataset:

####CREATE A COLUMN ID  
  
d %<>%  
 ungroup(d) %>%  
 mutate(id = 1 : 2230)  
d %<>%  
 mutate(total\_taxes = ifelse(d$total\_taxes < 1000, NA, total\_taxes))  
real\_y = data.frame(d$id, d$sale\_price)  
real\_d = subset(d, (!is.na(d$sale\_price)))  
fake\_d = subset(d, (is.na(d$sale\_price)))  
real\_d$sale\_price = NULL  
fake\_d$sale\_price = NULL

#Split the data that has y into train and test sets

train\_indices = sample(1 : nrow(real\_d), nrow(real\_d)\*4/5)  
training\_data = real\_d[train\_indices, ]  
testing\_data = real\_d[-train\_indices, ]  
  
X = rbind(training\_data, testing\_data, fake\_d)

#Let’s first create a matrix with columns that represents missingness

M = tbl\_df(apply(is.na(X), 2, as.numeric))  
colnames(M) = paste("is\_missing\_", colnames(X), sep = "")

#Some of these missing indicators are collinear because they share all the rows they are missing on. Let’s filter those out:

M = tbl\_df(t(unique(t(M))))

#Some featuers did not have missingness so let’s remove them:

M %<>% select\_if(function(x){sum(x) > 0})

Now let’s impute using the package. we cannot fit RF models to the entire dataset (it’s 26,000! observations) so we will sample 5 for X1 and for each of the trees and then average. That will be good enough.

pacman::p\_load(missForest)  
Ximp = missForest(data.frame(X), sampsize = rep(172, ncol(X)))$ximp

## missForest iteration 1 in progress...

## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =  
## mtry, : The response has five or fewer unique values. Are you sure you want  
## to do regression?

## done!  
## missForest iteration 2 in progress...

## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =  
## mtry, : The response has five or fewer unique values. Are you sure you want  
## to do regression?

## done!  
## missForest iteration 3 in progress...

## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =  
## mtry, : The response has five or fewer unique values. Are you sure you want  
## to do regression?

## done!  
## missForest iteration 4 in progress...

## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =  
## mtry, : The response has five or fewer unique values. Are you sure you want  
## to do regression?

## done!

Ximp %<>%  
 arrange(id)  
  
Xnew = data.frame(cbind(Ximp, M, real\_y))  
  
Xnew %<>%  
 mutate(price = d.sale\_price) %>%  
 select(-c(id, d.id, d.sale\_price))  
   
linear\_mod\_impute\_and\_missing\_dummies = lm(price ~ ., data = Xnew)  
summary(linear\_mod\_impute\_and\_missing\_dummies)

##   
## Call:  
## lm(formula = price ~ ., data = Xnew)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -296854 -36454 -20 36523 332799   
##   
## Coefficients: (3 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -3.664e+07 9.154e+06 -4.003 7.23e-05  
## approx\_year\_built 3.249e+01 2.453e+02 0.132 0.894659  
## community\_district\_num 2.337e+03 1.169e+03 2.000 0.046099  
## coop\_condocondo 1.419e+05 1.729e+04 8.208 1.98e-15  
## dining\_room\_typedining area 1.559e+04 5.259e+04 0.297 0.766940  
## dining\_room\_typeformal 2.332e+04 8.519e+03 2.738 0.006405  
## dining\_room\_typeother 1.297e+04 1.126e+04 1.152 0.249739  
## garage\_exists 5.128e+02 9.017e+03 0.057 0.954675  
## kitchen\_typeeat in -2.854e+03 1.000e+04 -0.285 0.775535  
## kitchen\_typeefficiency -1.767e+04 9.750e+03 -1.813 0.070508  
## num\_bedrooms 4.236e+04 7.904e+03 5.359 1.29e-07  
## num\_floors\_in\_building 2.535e+03 7.163e+02 3.539 0.000440  
## num\_full\_bathrooms 1.727e+04 5.242e+04 0.329 0.741999  
## num\_half\_bathrooms -1.230e+04 3.258e+04 -0.377 0.706010  
## num\_total\_rooms 1.853e+04 5.175e+03 3.581 0.000376  
## parking\_charges 3.493e+02 9.717e+01 3.595 0.000357  
## pct\_tax\_deductibl 4.671e+01 1.026e+03 0.046 0.963700  
## sq\_footage 3.027e+01 1.268e+01 2.386 0.017398  
## total\_taxes 1.314e+01 5.344e+00 2.458 0.014296  
## walk\_score -6.554e+02 3.382e+02 -1.938 0.053253  
## lat 6.530e+05 1.354e+05 4.822 1.90e-06  
## lon -1.301e+05 8.466e+04 -1.537 0.125057  
## pets\_allowed 1.112e+04 6.886e+03 1.615 0.106929  
## monthly\_cost 1.286e+02 1.416e+01 9.080 < 2e-16  
## price\_persqft 5.546e+05 6.661e+04 8.326 8.36e-16  
## is\_missing\_approx\_year\_built 6.389e+03 3.368e+04 0.190 0.849625  
## is\_missing\_community\_district\_num -2.312e+05 7.469e+04 -3.095 0.002080  
## is\_missing\_dining\_room\_type 5.110e+03 7.818e+03 0.654 0.513701  
## is\_missing\_kitchen\_type -1.343e+04 2.871e+04 -0.468 0.640234  
## is\_missing\_num\_bedrooms NA NA NA NA  
## is\_missing\_num\_floors\_in\_building 2.070e+03 8.315e+03 0.249 0.803508  
## is\_missing\_num\_half\_bathrooms -1.221e+04 1.407e+04 -0.867 0.386093  
## is\_missing\_num\_total\_rooms NA NA NA NA  
## is\_missing\_parking\_charges -2.146e+03 7.632e+03 -0.281 0.778657  
## is\_missing\_pct\_tax\_deductibl 1.291e+04 8.669e+03 1.489 0.137226  
## is\_missing\_sq\_footage -2.267e+03 6.700e+03 -0.338 0.735281  
## is\_missing\_total\_taxes 6.496e+02 9.155e+03 0.071 0.943463  
## is\_missing\_monthly\_cost -2.973e+03 1.991e+04 -0.149 0.881388  
## is\_missing\_price\_persqft NA NA NA NA  
##   
## (Intercept) \*\*\*  
## approx\_year\_built   
## community\_district\_num \*   
## coop\_condocondo \*\*\*  
## dining\_room\_typedining area   
## dining\_room\_typeformal \*\*   
## dining\_room\_typeother   
## garage\_exists   
## kitchen\_typeeat in   
## kitchen\_typeefficiency .   
## num\_bedrooms \*\*\*  
## num\_floors\_in\_building \*\*\*  
## num\_full\_bathrooms   
## num\_half\_bathrooms   
## num\_total\_rooms \*\*\*  
## parking\_charges \*\*\*  
## pct\_tax\_deductibl   
## sq\_footage \*   
## total\_taxes \*   
## walk\_score .   
## lat \*\*\*  
## lon   
## pets\_allowed   
## monthly\_cost \*\*\*  
## price\_persqft \*\*\*  
## is\_missing\_approx\_year\_built   
## is\_missing\_community\_district\_num \*\*   
## is\_missing\_dining\_room\_type   
## is\_missing\_kitchen\_type   
## is\_missing\_num\_bedrooms   
## is\_missing\_num\_floors\_in\_building   
## is\_missing\_num\_half\_bathrooms   
## is\_missing\_num\_total\_rooms   
## is\_missing\_parking\_charges   
## is\_missing\_pct\_tax\_deductibl   
## is\_missing\_sq\_footage   
## is\_missing\_total\_taxes   
## is\_missing\_monthly\_cost   
## is\_missing\_price\_persqft   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 72110 on 492 degrees of freedom  
## (1702 observations deleted due to missingness)  
## Multiple R-squared: 0.8494, Adjusted R-squared: 0.8387   
## F-statistic: 79.28 on 35 and 492 DF, p-value: < 2.2e-16

### REMOVING MISSING Y SECTION

Data = Xnew  
### sale price is our imputed Y  
  
Y = Data$price  
  
Data %<>%  
 filter(!is.na(price)) %>%  
 select(-price)  
  
  
  
  
Xtrain = Data[1:422, ]  
Xtest = Data[423:528, ]  
  
Ytrain = Y[1:422]  
Ytest = Y[423:528]  
  
dtrain = cbind(Xtrain, Ytrain) ## combine x train with y train, x test with y test  
dtest = cbind(Xtest, Ytest)

## Dropping colinear features

Xtrain %<>%  
 select(-c(is\_missing\_num\_total\_rooms, is\_missing\_num\_bedrooms, is\_missing\_price\_persqft))

Linear Regression

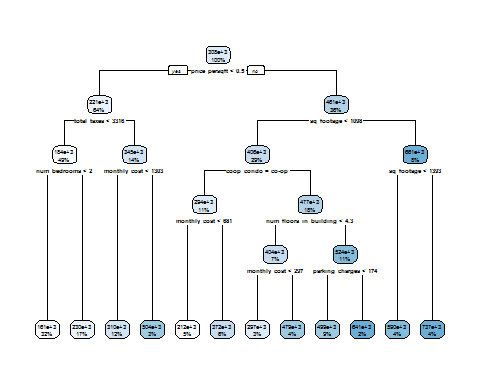
linear = lm(Ytrain ~ ., data = Xtrain)## simple linear model  
summary(linear)

##   
## Call:  
## lm(formula = Ytrain ~ ., data = Xtrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -295653 -31899 2725 33423 314214   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -3.794e+07 1.017e+07 -3.730 0.000220  
## approx\_year\_built -3.775e+01 2.690e+02 -0.140 0.888450  
## community\_district\_num 2.001e+03 1.243e+03 1.610 0.108301  
## coop\_condocondo 1.720e+05 1.977e+04 8.703 < 2e-16  
## dining\_room\_typedining area 1.950e+04 5.172e+04 0.377 0.706350  
## dining\_room\_typeformal 2.424e+04 9.331e+03 2.598 0.009733  
## dining\_room\_typeother 1.246e+04 1.221e+04 1.020 0.308268  
## garage\_exists 3.292e+03 1.023e+04 0.322 0.747841  
## kitchen\_typeeat in 5.448e+03 1.092e+04 0.499 0.618207  
## kitchen\_typeefficiency -1.649e+04 1.078e+04 -1.530 0.126950  
## num\_bedrooms 3.235e+04 8.779e+03 3.685 0.000261  
## num\_floors\_in\_building 2.363e+03 7.932e+02 2.980 0.003069  
## num\_full\_bathrooms 1.706e+04 5.152e+04 0.331 0.740723  
## num\_half\_bathrooms 2.510e+03 3.594e+04 0.070 0.944363  
## num\_total\_rooms 2.179e+04 5.748e+03 3.791 0.000174  
## parking\_charges 3.950e+02 1.032e+02 3.827 0.000151  
## pct\_tax\_deductibl 5.616e+01 1.417e+03 0.040 0.968404  
## sq\_footage 2.578e+01 1.316e+01 1.958 0.050927  
## total\_taxes 7.102e+00 5.832e+00 1.218 0.224030  
## walk\_score -5.797e+02 3.738e+02 -1.551 0.121804  
## lat 6.779e+05 1.472e+05 4.606 5.59e-06  
## lon -1.359e+05 9.529e+04 -1.426 0.154661  
## pets\_allowed 5.998e+03 7.596e+03 0.790 0.430213  
## monthly\_cost 1.630e+02 1.876e+01 8.688 < 2e-16  
## price\_persqft 4.754e+05 7.376e+04 6.445 3.44e-10  
## is\_missing\_approx\_year\_built 1.324e+04 3.309e+04 0.400 0.689335  
## is\_missing\_community\_district\_num -2.695e+05 7.382e+04 -3.651 0.000297  
## is\_missing\_dining\_room\_type 5.126e+02 8.653e+03 0.059 0.952789  
## is\_missing\_kitchen\_type -5.278e+03 2.832e+04 -0.186 0.852252  
## is\_missing\_num\_floors\_in\_building 6.807e+02 9.305e+03 0.073 0.941719  
## is\_missing\_num\_half\_bathrooms 8.973e+02 1.691e+04 0.053 0.957718  
## is\_missing\_parking\_charges -5.735e+03 8.255e+03 -0.695 0.487666  
## is\_missing\_pct\_tax\_deductibl 1.016e+04 9.496e+03 1.070 0.285327  
## is\_missing\_sq\_footage 2.216e+03 7.544e+03 0.294 0.769094  
## is\_missing\_total\_taxes 6.810e+02 1.000e+04 0.068 0.945737  
## is\_missing\_monthly\_cost -7.148e+03 2.288e+04 -0.312 0.754886  
##   
## (Intercept) \*\*\*  
## approx\_year\_built   
## community\_district\_num   
## coop\_condocondo \*\*\*  
## dining\_room\_typedining area   
## dining\_room\_typeformal \*\*   
## dining\_room\_typeother   
## garage\_exists   
## kitchen\_typeeat in   
## kitchen\_typeefficiency   
## num\_bedrooms \*\*\*  
## num\_floors\_in\_building \*\*   
## num\_full\_bathrooms   
## num\_half\_bathrooms   
## num\_total\_rooms \*\*\*  
## parking\_charges \*\*\*  
## pct\_tax\_deductibl   
## sq\_footage .   
## total\_taxes   
## walk\_score   
## lat \*\*\*  
## lon   
## pets\_allowed   
## monthly\_cost \*\*\*  
## price\_persqft \*\*\*  
## is\_missing\_approx\_year\_built   
## is\_missing\_community\_district\_num \*\*\*  
## is\_missing\_dining\_room\_type   
## is\_missing\_kitchen\_type   
## is\_missing\_num\_floors\_in\_building   
## is\_missing\_num\_half\_bathrooms   
## is\_missing\_parking\_charges   
## is\_missing\_pct\_tax\_deductibl   
## is\_missing\_sq\_footage   
## is\_missing\_total\_taxes   
## is\_missing\_monthly\_cost   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 70340 on 386 degrees of freedom  
## Multiple R-squared: 0.855, Adjusted R-squared: 0.8418   
## F-statistic: 65.03 on 35 and 386 DF, p-value: < 2.2e-16

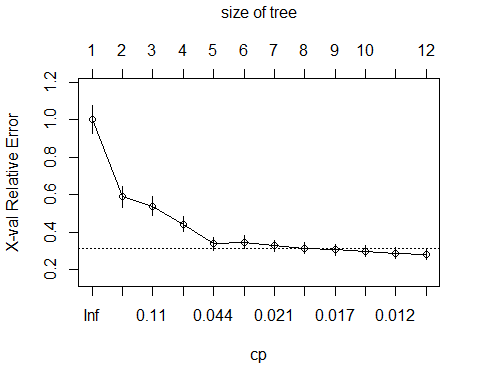
yhat = predict(linear, Xtest)  
  
e = yhat - Ytest  
  
sqrt(sum(e^2) / nrow(Xtest))

## [1] 83215.95

#REGRESSION TREE  
  
pacman::p\_load(rsample)#data spliting  
pacman::p\_load(rpart) #performing reg tree  
pacman::p\_load(rpart.plot) #ploting reg tree  
pacman::p\_load(ipred) #bagging  
pacman::p\_load(caret) #bagging  
  
  
  
m1 = rpart(  
 formula = Ytrain ~ .,  
 data = Xtrain,  
 method = "anova"  
 )  
  
rpart.plot(m1)



plotcp(m1)



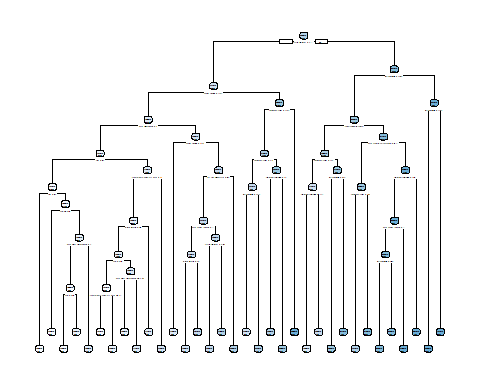
summary(m1)

## Call:  
## rpart(formula = Ytrain ~ ., data = Xtrain, method = "anova")  
## n= 422   
##   
## CP nsplit rel error xerror xstd  
## 1 0.42758149 0 1.0000000 1.0034864 0.07563004  
## 2 0.12824221 1 0.5724185 0.5888809 0.05677089  
## 3 0.09266116 2 0.4441763 0.5377683 0.05130348  
## 4 0.07319492 3 0.3515151 0.4417798 0.04077420  
## 5 0.02585520 4 0.2783202 0.3364894 0.03348648  
## 6 0.02291037 5 0.2524650 0.3462884 0.03336475  
## 7 0.01939333 6 0.2295546 0.3283003 0.02936593  
## 8 0.01776872 7 0.2101613 0.3146267 0.03040719  
## 9 0.01663924 8 0.1923926 0.3055096 0.02989080  
## 10 0.01348814 9 0.1757533 0.2983826 0.03004550  
## 11 0.01014447 10 0.1622652 0.2876490 0.02931209  
## 12 0.01000000 11 0.1521207 0.2813368 0.02881401  
##   
## Variable importance  
## price\_persqft monthly\_cost coop\_condo   
## 20 15 13   
## approx\_year\_built sq\_footage total\_taxes   
## 12 9 8   
## parking\_charges lon num\_bedrooms   
## 8 5 3   
## num\_total\_rooms pct\_tax\_deductibl community\_district\_num   
## 3 2 1   
## num\_floors\_in\_building lat   
## 1 1   
##   
## Node number 1: 422 observations, complexity param=0.4275815  
## mean=308191.7, MSE=3.121006e+10   
## left son=2 (268 obs) right son=3 (154 obs)  
## Primary splits:  
## price\_persqft < 0.5036155 to the left, improve=0.4275815, (0 missing)  
## coop\_condo splits as LR, improve=0.3754617, (0 missing)  
## approx\_year\_built < 1970.5 to the left, improve=0.3463094, (0 missing)  
## total\_taxes < 3389.792 to the left, improve=0.3270625, (0 missing)  
## num\_total\_rooms < 4.5 to the left, improve=0.2781774, (0 missing)  
## Surrogate splits:  
## coop\_condo splits as LR, agree=0.848, adj=0.584, (0 split)  
## approx\_year\_built < 1970.5 to the left, agree=0.844, adj=0.571, (0 split)  
## parking\_charges < 135.1167 to the left, agree=0.794, adj=0.435, (0 split)  
## monthly\_cost < 471.5 to the right, agree=0.789, adj=0.422, (0 split)  
## lon < -73.87994 to the right, agree=0.744, adj=0.299, (0 split)  
##   
## Node number 2: 268 observations, complexity param=0.09266116  
## mean=220622.8, MSE=1.036057e+10   
## left son=4 (207 obs) right son=5 (61 obs)  
## Primary splits:  
## total\_taxes < 3316.375 to the left, improve=0.4395279, (0 missing)  
## sq\_footage < 940.6025 to the left, improve=0.4138876, (0 missing)  
## monthly\_cost < 1019 to the left, improve=0.3937581, (0 missing)  
## num\_total\_rooms < 4.5 to the left, improve=0.3399684, (0 missing)  
## num\_bedrooms < 1.5 to the left, improve=0.3103121, (0 missing)  
## Surrogate splits:  
## sq\_footage < 1050.357 to the left, agree=0.858, adj=0.377, (0 split)  
## monthly\_cost < 985.5 to the left, agree=0.851, adj=0.344, (0 split)  
## lat < 40.77906 to the left, agree=0.832, adj=0.262, (0 split)  
## num\_total\_rooms < 5.5 to the left, agree=0.810, adj=0.164, (0 split)  
## num\_floors\_in\_building < 7.36 to the left, agree=0.806, adj=0.148, (0 split)  
##   
## Node number 3: 154 observations, complexity param=0.1282422  
## mean=460584.3, MSE=3.092526e+10   
## left son=6 (121 obs) right son=7 (33 obs)  
## Primary splits:  
## sq\_footage < 1098 to the left, improve=0.3546534, (0 missing)  
## num\_total\_rooms < 4.5 to the left, improve=0.3330176, (0 missing)  
## num\_bedrooms < 1.5 to the left, improve=0.3201549, (0 missing)  
## total\_taxes < 3690.024 to the left, improve=0.3156214, (0 missing)  
## approx\_year\_built < 1963.5 to the left, improve=0.2232692, (0 missing)  
## Surrogate splits:  
## total\_taxes < 4120.488 to the left, agree=0.890, adj=0.485, (0 split)  
## num\_bedrooms < 2.5 to the left, agree=0.857, adj=0.333, (0 split)  
## monthly\_cost < 1478.5 to the left, agree=0.844, adj=0.273, (0 split)  
## num\_total\_rooms < 5.5 to the left, agree=0.805, adj=0.091, (0 split)  
## pct\_tax\_deductibl < 35.53167 to the right, agree=0.799, adj=0.061, (0 split)  
##   
## Node number 4: 207 observations, complexity param=0.01663924  
## mean=183990.5, MSE=3.585262e+09   
## left son=8 (137 obs) right son=9 (70 obs)  
## Primary splits:  
## num\_bedrooms < 1.5 to the left, improve=0.2952904, (0 missing)  
## monthly\_cost < 764 to the left, improve=0.2228783, (0 missing)  
## sq\_footage < 940.6025 to the left, improve=0.2185199, (0 missing)  
## total\_taxes < 2953.945 to the left, improve=0.2160151, (0 missing)  
## num\_total\_rooms < 4.5 to the left, improve=0.2120814, (0 missing)  
## Surrogate splits:  
## sq\_footage < 853 to the left, agree=0.874, adj=0.629, (0 split)  
## num\_total\_rooms < 3.5 to the left, agree=0.845, adj=0.543, (0 split)  
## monthly\_cost < 805.5 to the left, agree=0.773, adj=0.329, (0 split)  
## total\_taxes < 3140.35 to the left, agree=0.744, adj=0.243, (0 split)  
## num\_half\_bathrooms < 0.965 to the left, agree=0.691, adj=0.086, (0 split)  
##   
## Node number 5: 61 observations, complexity param=0.0258552  
## mean=344932.6, MSE=1.334551e+10   
## left son=10 (50 obs) right son=11 (11 obs)  
## Primary splits:  
## monthly\_cost < 1302.5 to the left, improve=0.4183022, (0 missing)  
## sq\_footage < 1102.955 to the left, improve=0.3724499, (0 missing)  
## total\_taxes < 4024.79 to the left, improve=0.2772907, (0 missing)  
## price\_persqft < 0.4499131 to the left, improve=0.2409971, (0 missing)  
## parking\_charges < 64.775 to the left, improve=0.2147612, (0 missing)  
## Surrogate splits:  
## num\_total\_rooms < 6.5 to the left, agree=0.885, adj=0.364, (0 split)  
## sq\_footage < 1339.5 to the left, agree=0.869, adj=0.273, (0 split)  
## total\_taxes < 4381.68 to the left, agree=0.852, adj=0.182, (0 split)  
## num\_half\_bathrooms < 0.72 to the right, agree=0.836, adj=0.091, (0 split)  
##   
## Node number 6: 121 observations, complexity param=0.07319492  
## mean=405892.4, MSE=2.159689e+10   
## left son=12 (47 obs) right son=13 (74 obs)  
## Primary splits:  
## coop\_condo splits as LR, improve=0.3689025, (0 missing)  
## approx\_year\_built < 2004.5 to the left, improve=0.2996678, (0 missing)  
## price\_persqft < 0.5847979 to the left, improve=0.2978430, (0 missing)  
## num\_total\_rooms < 3.5 to the left, improve=0.2590936, (0 missing)  
## sq\_footage < 679.415 to the left, improve=0.2478039, (0 missing)  
## Surrogate splits:  
## approx\_year\_built < 1971 to the left, agree=0.884, adj=0.702, (0 split)  
## monthly\_cost < 514.5 to the right, agree=0.851, adj=0.617, (0 split)  
## price\_persqft < 0.6130331 to the left, agree=0.818, adj=0.532, (0 split)  
## community\_district\_num < 27.5 to the right, agree=0.769, adj=0.404, (0 split)  
## pct\_tax\_deductibl < 48.45333 to the right, agree=0.752, adj=0.362, (0 split)  
##   
## Node number 7: 33 observations, complexity param=0.01348814  
## mean=661121.2, MSE=1.394647e+10   
## left son=14 (17 obs) right son=15 (16 obs)  
## Primary splits:  
## sq\_footage < 1392.81 to the left, improve=0.3859944, (0 missing)  
## total\_taxes < 3874.715 to the left, improve=0.3050610, (0 missing)  
## monthly\_cost < 1326 to the left, improve=0.2652581, (0 missing)  
## num\_bedrooms < 2.5 to the left, improve=0.2378804, (0 missing)  
## num\_floors\_in\_building < 21.5 to the left, improve=0.1998063, (0 missing)  
## Surrogate splits:  
## total\_taxes < 4366.515 to the left, agree=0.818, adj=0.625, (0 split)  
## monthly\_cost < 1326 to the left, agree=0.788, adj=0.562, (0 split)  
## num\_bedrooms < 2.5 to the left, agree=0.727, adj=0.438, (0 split)  
## lon < -73.83932 to the right, agree=0.727, adj=0.438, (0 split)  
## price\_persqft < 0.5384526 to the right, agree=0.727, adj=0.438, (0 split)  
##   
## Node number 8: 137 observations  
## mean=160732.4, MSE=1.951881e+09   
##   
## Node number 9: 70 observations  
## mean=229509.8, MSE=3.651313e+09   
##   
## Node number 10: 50 observations  
## mean=309887.8, MSE=7.481011e+09   
##   
## Node number 11: 11 observations  
## mean=504227.3, MSE=9.045062e+09   
##   
## Node number 12: 47 observations, complexity param=0.02291037  
## mean=293892.3, MSE=1.181581e+10   
## left son=24 (23 obs) right son=25 (24 obs)  
## Primary splits:  
## monthly\_cost < 681 to the left, improve=0.5433477, (0 missing)  
## sq\_footage < 743.825 to the left, improve=0.5232991, (0 missing)  
## total\_taxes < 2752.269 to the left, improve=0.4687540, (0 missing)  
## num\_total\_rooms < 3.5 to the left, improve=0.4296906, (0 missing)  
## num\_bedrooms < 0.5 to the left, improve=0.3338656, (0 missing)  
## Surrogate splits:  
## sq\_footage < 743.825 to the left, agree=0.894, adj=0.783, (0 split)  
## total\_taxes < 3129.924 to the left, agree=0.872, adj=0.739, (0 split)  
## num\_total\_rooms < 3.5 to the left, agree=0.766, adj=0.522, (0 split)  
## price\_persqft < 0.5539046 to the left, agree=0.745, adj=0.478, (0 split)  
## pct\_tax\_deductibl < 43.415 to the right, agree=0.723, adj=0.435, (0 split)  
##   
## Node number 13: 74 observations, complexity param=0.01939333  
## mean=477027.5, MSE=1.478183e+10   
## left son=26 (29 obs) right son=27 (45 obs)  
## Primary splits:  
## num\_floors\_in\_building < 4.285 to the left, improve=0.2335069, (0 missing)  
## monthly\_cost < 211 to the left, improve=0.2193306, (0 missing)  
## total\_taxes < 2143 to the left, improve=0.2122442, (0 missing)  
## approx\_year\_built < 2006.5 to the left, improve=0.2079009, (0 missing)  
## parking\_charges < 174.5133 to the left, improve=0.1971597, (0 missing)  
## Surrogate splits:  
## price\_persqft < 0.5718783 to the left, agree=0.730, adj=0.310, (0 split)  
## total\_taxes < 2143 to the left, agree=0.716, adj=0.276, (0 split)  
## parking\_charges < 114.89 to the left, agree=0.703, adj=0.241, (0 split)  
## community\_district\_num < 24.5 to the left, agree=0.676, adj=0.172, (0 split)  
## pct\_tax\_deductibl < 46.875 to the right, agree=0.676, adj=0.172, (0 split)  
##   
## Node number 14: 17 observations  
## mean=589941.2, MSE=3.455467e+09   
##   
## Node number 15: 16 observations  
## mean=736750, MSE=1.399019e+10   
##   
## Node number 24: 23 observations  
## mean=212043.5, MSE=3.314759e+09   
##   
## Node number 25: 24 observations  
## mean=372330.8, MSE=7.389972e+09   
##   
## Node number 26: 29 observations, complexity param=0.01776872  
## mean=403842.7, MSE=1.358682e+10   
## left son=52 (12 obs) right son=53 (17 obs)  
## Primary splits:  
## monthly\_cost < 297 to the left, improve=0.5939465, (0 missing)  
## kitchen\_type splits as LRL-, improve=0.2566576, (0 missing)  
## total\_taxes < 2087.5 to the left, improve=0.2457086, (0 missing)  
## sq\_footage < 698.275 to the left, improve=0.2435420, (0 missing)  
## num\_bedrooms < 1.5 to the left, improve=0.2292115, (0 missing)  
## Surrogate splits:  
## num\_bedrooms < 1.5 to the left, agree=0.793, adj=0.500, (0 split)  
## num\_total\_rooms < 3.5 to the left, agree=0.793, adj=0.500, (0 split)  
## sq\_footage < 669.49 to the left, agree=0.793, adj=0.500, (0 split)  
## price\_persqft < 0.6518208 to the right, agree=0.793, adj=0.500, (0 split)  
## total\_taxes < 2306.5 to the left, agree=0.759, adj=0.417, (0 split)  
##   
## Node number 27: 45 observations, complexity param=0.01014447  
## mean=524191.1, MSE=9.875883e+09   
## left son=54 (37 obs) right son=55 (8 obs)  
## Primary splits:  
## parking\_charges < 174.1833 to the left, improve=0.3006410, (0 missing)  
## monthly\_cost < 630.5 to the left, improve=0.2754588, (0 missing)  
## approx\_year\_built < 2005.5 to the left, improve=0.2286395, (0 missing)  
## total\_taxes < 3674.215 to the left, improve=0.1941959, (0 missing)  
## sq\_footage < 878.5 to the left, improve=0.1876615, (0 missing)  
## Surrogate splits:  
## price\_persqft < 0.8290999 to the left, agree=0.956, adj=0.750, (0 split)  
## lon < -73.92553 to the right, agree=0.933, adj=0.625, (0 split)  
## approx\_year\_built < 2009.5 to the left, agree=0.889, adj=0.375, (0 split)  
## community\_district\_num < 29.5 to the left, agree=0.867, adj=0.250, (0 split)  
## sq\_footage < 591 to the right, agree=0.867, adj=0.250, (0 split)  
##   
## Node number 52: 12 observations  
## mean=296920.8, MSE=1.016437e+10   
##   
## Node number 53: 17 observations  
## mean=479316.9, MSE=2.236462e+09   
##   
## Node number 54: 37 observations  
## mean=498854.1, MSE=6.488097e+09   
##   
## Node number 55: 8 observations  
## mean=641375, MSE=8.843234e+09

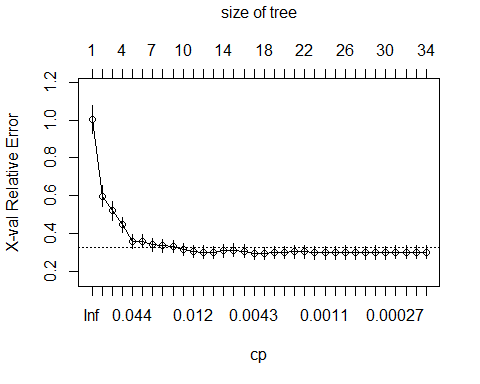
yhat = predict(m1, Xtest)  
e = yhat - Ytest  
sqrt(sum(e^2)/106)

## [1] 111454.6

m2 <- rpart(  
 formula = Ytrain ~ .,  
 data = Xtrain,  
 method = "anova",   
 control = list(cp = 0, xval = 10)  
)  
  
rpart.plot(m2)



plotcp(m2)



yhat = predict(m2, Xtest)  
e = yhat - Ytest  
sqrt(sum(e^2)/106)

## [1] 107028.4

jpeg(file = "save\_m2.jpeg")

###Tuning  
m3 <- rpart(  
 formula = Ytrain ~ .,  
 data = Xtrain,  
 method = "anova",   
 control = list(minsplit = 10, maxdepth = 12, xval = 10)  
)  
  
yhat = predict(m3, Xtest)  
e = yhat - Ytest  
sqrt(sum(e^2)/106)

## [1] 111454.6

m3$cptable

## CP nsplit rel error xerror xstd  
## 1 0.42758149 0 1.0000000 1.0033581 0.07572629  
## 2 0.12824221 1 0.5724185 0.6290791 0.05820362  
## 3 0.09266116 2 0.4441763 0.5672022 0.05401394  
## 4 0.07319492 3 0.3515151 0.4550638 0.04217392  
## 5 0.02585520 4 0.2783202 0.3534294 0.03651200  
## 6 0.02291037 5 0.2524650 0.3043492 0.02887121  
## 7 0.01939333 6 0.2295546 0.3147219 0.02938829  
## 8 0.01776872 7 0.2101613 0.3044802 0.02975724  
## 9 0.01663924 8 0.1923926 0.3003660 0.02986459  
## 10 0.01348814 9 0.1757533 0.2768037 0.02751636  
## 11 0.01014447 10 0.1622652 0.2655842 0.02747896  
## 12 0.01000000 11 0.1521207 0.2619472 0.02733595

# function to get optimal cp  
get\_cp <- function(x) {  
 min <- which.min(x$cptable[, "xerror"])  
 cp <- x$cptable[min, "CP"]   
}  
  
# function to get minimum error  
get\_min\_error <- function(x) {  
 min <- which.min(x$cptable[, "xerror"])  
 xerror <- x$cptable[min, "xerror"]   
}

optimal\_tree <- rpart(  
 formula = Ytrain ~ .,  
 data = Xtrain,  
 method = "anova",  
 control = list(minsplit = 11, maxdepth = 8, cp = 0.01)  
 )  
  
pred <- predict(optimal\_tree, newdata = Xtrain)  
RMSE(pred = pred, obs = Ytrain)

## [1] 68903.54

##RANDOM FORESTS

m1 <- randomForest(  
 formula = Ytrain ~ .,  
 data = Xtrain  
)  
  
m1

##   
## Call:  
## randomForest(formula = Ytrain ~ ., data = Xtrain)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 10  
##   
## Mean of squared residuals: 4408749600  
## % Var explained: 85.87

which.min(m1$mse)

## [1] 159

# RMSE of this optimal random forest  
sqrt(m1$mse[which.min(m1$mse)])

## [1] 65665.63

features <- setdiff(names(Xtrain), Ytrain)  
  
set.seed(1989)  
  
m2 <- tuneRF(  
 x = Xtrain,  
 y = Ytrain,  
 ntreeTry = 500,  
 mtryStart = 5,  
 stepFactor = 1.5,  
 improve = 0.01,  
 trace = FALSE # to not show real-time progress   
)

## -0.03910721 0.01   
## 0.03347455 0.01   
## 0.03056411 0.01   
## 0.01855087 0.01   
## 0.008980753 0.01