Citizen Science Efficacy Analysis

William Norfolk

11/6/2019

This script measures the efficacy of citizen science data gathered in this study by investigating the individual water quality parameters that characterize the seperation of observations into the selected outcome variable island\_side. Selection of island side will be evaluated using tree-based modeling to detemrine which water quality parameters strongly influence the the seperation of water systems. The goal of this analysis is not to determine which parameters are the most powerful for influencing data seperation, but rather to measure the most powerful variables and compare them to known conditions of water quality seperation. If the tree analysis of our citizen science data produces similar keystone predictor variables to our known strong predictors of water quality this will suggest that the citizen science data used is consistent with the actual condition of Key Largo waters thus showing providing a measure of efficacy for the usage of this citizen science data.

## Model Prep

Start by loading the libraries.

library('tidyr')  
library('dplyr')

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library('forcats')  
library('ggplot2')  
library('knitr')  
library('caret')

## Loading required package: lattice

library('doParallel')

## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel

library('rpart')  
library('rpart.plot')  
library('mda')

## Loading required package: class

## Loaded mda 0.4-10

library('ranger')  
library('e1071')  
library('visdat')  
library('filesstrings')

## Loading required package: stringr

##   
## Attaching package: 'filesstrings'

## The following object is masked from 'package:dplyr':  
##   
## all\_equal

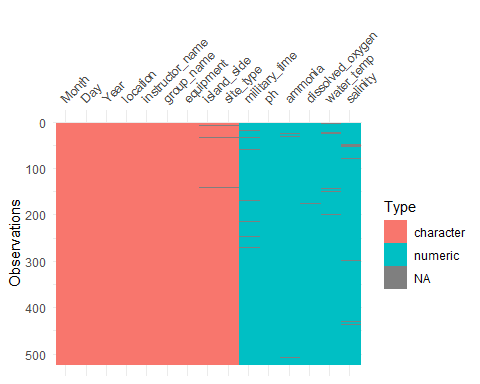
Load the data and take a look.

WQ\_clean\_data <- readRDS("../../data/processed\_data/processeddata.rds")  
  
glimpse(WQ\_clean\_data)

## Observations: 522  
## Variables: 15  
## $ Month <chr> "01", "02", "02", "02", "02", "02", "02", "02...  
## $ Day <chr> "08", "08", "08", "08", "09", "09", "09", "12...  
## $ Year <chr> "16", "16", "16", "16", "16", "16", "16", "16...  
## $ military\_time <dbl> 1415, 1515, 1550, 1555, 1001, 1015, 1022, 103...  
## $ location <chr> "Boat Ramp", "Grecian Dry Rocks", "Grecian Dr...  
## $ instructor\_name <chr> "Katy, Sarah, Driver", "Chelsea", "Katy, Tomm...  
## $ group\_name <chr> "NA", "McLean High School", "McLean High Scho...  
## $ ph <dbl> 8.0, 8.4, 8.2, 8.4, 8.0, 8.0, 8.0, 8.0, 8.4, ...  
## $ ammonia <dbl> 0.00, 0.00, 0.00, 0.00, 0.25, 0.00, 0.00, 0.0...  
## $ dissolved\_oxygen <dbl> 5.0, 4.0, 4.0, 6.0, 8.0, 4.0, 5.0, 6.0, 6.0, ...  
## $ water\_temp <dbl> NA, 23.5, 21.0, 36.0, 18.0, 18.0, 18.0, 18.3,...  
## $ salinity <dbl> 36, 40, 44, 35, 33, 30, 33, 35, 40, 30, 35, 2...  
## $ equipment <chr> "kit", "kit", "kit", "kit", "kit", "kit", "ki...  
## $ island\_side <chr> "ocean", "ocean", "ocean", "ocean", NA, "bay"...  
## $ site\_type <chr> "Seagrass/Mangrove", "Coral Reef", "Coral Ree...

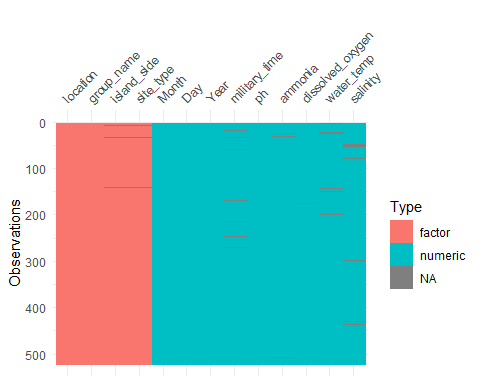
Check for NAs and properly formatted classes.

vis\_dat(WQ\_clean\_data)

 Data looks pretty good, there are some NA values but the tree models should be able to handel these since they are not strongly grouped.

We do need to recode a few variables in this view. We will make all or date related variables numeric, and the remaining buckets will be factors. Also we can remove instructor\_name and kit since they are not very relevent to the data. For now we will leave group\_name in the data in case different schools are more effective measurers of water quality.

#Drop instructor name and equipment  
reduce\_vars <- WQ\_clean\_data %>% dplyr::select(-instructor\_name, -equipment)  
  
#Change numerics  
reduce\_vars$Month <- as.numeric(as.character(reduce\_vars$Month))  
reduce\_vars$Day <- as.numeric(as.character(reduce\_vars$Day))  
reduce\_vars$Year <- as.numeric(as.character(reduce\_vars$Year))  
  
#Change factors  
reduce\_vars$group\_name <- as.factor(as.character(reduce\_vars$group\_name))  
reduce\_vars$island\_side <- as.factor(as.character(reduce\_vars$island\_side))  
reduce\_vars$site\_type <- as.factor(as.character(reduce\_vars$site\_type))  
reduce\_vars$location <- as.factor(as.character(reduce\_vars$location))  
  
  
vis\_dat(reduce\_vars)



This data looks ready to use. We may make futher modifications downstream as models produce results.

## Data spllitting

Before we split we need to ensure the NAs are removed from the outcome and the outcome is in the first column of the dataframe.

model\_ready <- reduce\_vars[!is.na(reduce\_vars$island\_side), ]  
  
model\_ready <- model\_ready[, c(12,1,2,3,4,5,6,7,8,9,10,11,13)]

We will split our data for K-fold cross-validation into a 70/30 train and test set.

set.seed(123)  
  
trainset <- caret::createDataPartition(y = model\_ready$island\_side, p = 0.7, list = FALSE)  
  
#Extract observations/rows for training and assign to new variable  
data\_train = model\_ready[trainset,]   
  
#Same as above but for the test set  
data\_test = model\_ready[-trainset,]

data\_train <- na.omit(data\_train)  
data\_test <- na.omit(data\_test)

## Null Model

First we need to determine our null model. Since our outcome is a categorical predictor, a null model should always predict the most common category. Four our dataset the oceanside is the most frequently visited island\_side. We will use accuracy to measure how well our model predicts the island\_side given the five water quality parameters.

library(mlr) #We will jump between mlr and caret so call with :: at all times.

## Loading required package: ParamHelpers

##   
## Attaching package: 'mlr'

## The following object is masked from 'package:e1071':  
##   
## impute

## The following object is masked from 'package:caret':  
##   
## train

mlr::measureACC("ocean", model\_ready$island\_side)

## [1] 0.6237817

It looks like we have an accuracy of about 0.62 for our null lets see if we can do better.

Now lets see if we can do better with some models. We will start with single a single predictor model to determine accuracy for each predictor variable.

set.seed(1111) #To keep reproducibility  
  
outcomename = "island\_side" #Name of outcome of interest  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5) #Sets K-fold cross-validation.  
  
Npred <- ncol(data\_train)-1 #Sets predictors minus the outcome in column 1.  
  
resultmat <- data.frame(Variable = names(data\_train)[-1], Accuracy = rep(0,Npred)) #Takes results and moves to dataframe named resultmat, measure set to accuracy.  
  
for (n in 2:ncol(data\_train))  
{  
 fit1 <- caret::train( as.formula(paste(outcomename, "~",names(data\_train)[n])) , data = data\_train, method = "rpart", trControl = fitControl, na.action = na.pass, tuneLength = 10)   
resultmat[n-1,2]= max(fit1$results$Accuracy)   
} #Loop to all variables through prediction. Method rpart = tree algorithm, and skipping any NA values.   
  
print(resultmat)

## Variable Accuracy  
## 1 Month 0.6400157  
## 2 Day 0.6312873  
## 3 Year 0.6400184  
## 4 military\_time 0.6819280  
## 5 location 0.9064714  
## 6 group\_name 0.6640309  
## 7 ph 0.6208805  
## 8 ammonia 0.6276949  
## 9 dissolved\_oxygen 0.6553834  
## 10 water\_temp 0.6795003  
## 11 salinity 0.7324388  
## 12 site\_type 0.9293161

So the single-predictor looks pretty interesting. The majority of the predictors are around the same level as the null, however we do have some spikes. Location and site\_type have a very high accuracy. This is likely due to the fact that they are directly associated with the island\_side. We will likely have to remove these down the line since they are too powerful. Salinity is a bit stronger than the null as well, this is likely due to the large split in salinity from each island side we saw in the visualization. This is likely to become a very strong predictor in the model, and there is a possibility that salinity alone is enough to establish island\_side.

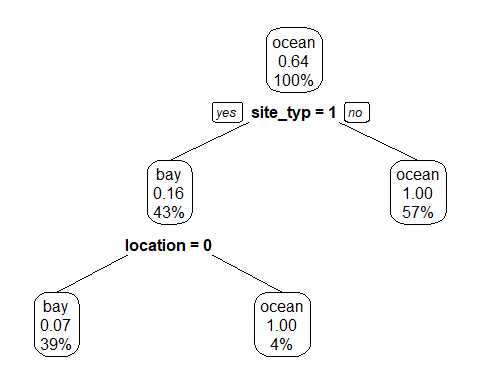
We will continue with a full model.

set.seed(1111)  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
  
fit1 = caret::train(island\_side ~ ., data=data\_train, method="rpart", trControl = fitControl, na.action = na.pass, tuneLength = 10)   
  
print(fit1$results)

## cp Accuracy Kappa AccuracySD KappaSD  
## 1 0.00000000 0.9612558 0.9161962 0.02038649 0.04563765  
## 2 0.08926876 0.9723048 0.9410796 0.01402703 0.02943670  
## 3 0.17853751 0.9291941 0.8531511 0.02045383 0.04102737  
## 4 0.26780627 0.9291941 0.8531511 0.02045383 0.04102737  
## 5 0.35707502 0.9291941 0.8531511 0.02045383 0.04102737  
## 6 0.44634378 0.9291941 0.8531511 0.02045383 0.04102737  
## 7 0.53561254 0.9291941 0.8531511 0.02045383 0.04102737  
## 8 0.62488129 0.9291941 0.8531511 0.02045383 0.04102737  
## 9 0.71415005 0.9291941 0.8531511 0.02045383 0.04102737  
## 10 0.80341880 0.7825070 0.4272646 0.13928909 0.41918687

Looks like we have pretty high accuracy across the board here. This is likely due to the two predictors mentioned above. Lets take a look at the tree.

prp(fit1$finalModel, extra = 106, type = 2)



As expected our tree splits along the lines of site\_type and location. This makes complete sense since island\_side and site\_type and essentially large subgroups of the various location information.To get a comprehensive picture of water qiality-based predictions we will remove these two variables from our data set.

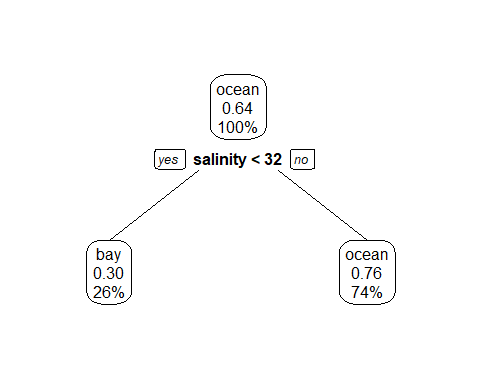
data\_train <- data\_train %>% dplyr::select(-location, -site\_type)  
data\_test <- data\_test %>% dplyr::select(-location, -site\_type)

Lets try the models again

set.seed(1111)   
outcome = "island\_side"  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
Npred <- ncol(data\_train)-1  
resultmat <- data.frame(Variable = names(data\_train)[-1], Accuracy = rep(0,Npred))  
for (n in 2:ncol(data\_train))  
{  
 fit\_single\_reduced <- caret::train( as.formula(paste(outcomename, "~",names(data\_train)[n])) , data = data\_train, method = "rpart", trControl = fitControl, na.action = na.pass, tuneLength = 20)   
resultmat[n-1,2]= max(fit\_single\_reduced$results$Accuracy)   
}  
  
print(resultmat)

## Variable Accuracy  
## 1 Month 0.6400157  
## 2 Day 0.6312873  
## 3 Year 0.6400184  
## 4 military\_time 0.6819280  
## 5 group\_name 0.6627287  
## 6 ph 0.6184522  
## 7 ammonia 0.6239860  
## 8 dissolved\_oxygen 0.6528715  
## 9 water\_temp 0.6794330  
## 10 salinity 0.7267083

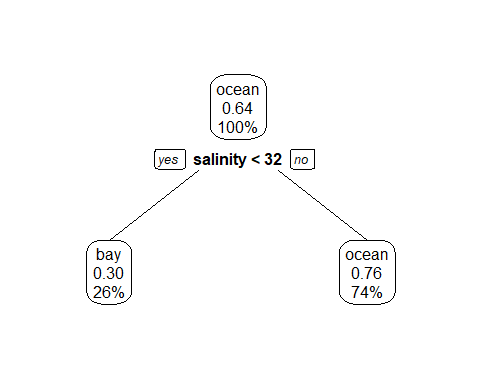
prp(fit\_single\_reduced$finalModel, extra = 106, type = 2)

 It looks like salinity is the strongest water quality predictor of island side in the dataset. Lets see if the full fit changes anything.

set.seed(1111)  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
  
fit2 = caret::train(island\_side ~ ., data=data\_train, method="rpart", trControl = fitControl, na.action = na.pass, tuneLength = 20)   
  
print(fit2$results)

## cp Accuracy Kappa AccuracySD KappaSD  
## 1 0.00000000 0.6916026 0.3095757 0.04115411 0.09967093  
## 2 0.01529465 0.7033153 0.3338135 0.04319221 0.10735552  
## 3 0.03058929 0.7174315 0.3685079 0.04632199 0.10999356  
## 4 0.04588394 0.7286049 0.3801122 0.04427078 0.10039975  
## 5 0.06117859 0.7292582 0.3848183 0.04476563 0.10439746  
## 6 0.07647323 0.7292582 0.3848183 0.04476563 0.10439746  
## 7 0.09176788 0.7292582 0.3848183 0.04476563 0.10439746  
## 8 0.10706253 0.7292582 0.3848183 0.04476563 0.10439746  
## 9 0.12235717 0.7292582 0.3848183 0.04476563 0.10439746  
## 10 0.13765182 0.7292582 0.3848183 0.04476563 0.10439746  
## 11 0.15294647 0.7292582 0.3848183 0.04476563 0.10439746  
## 12 0.16824112 0.7292582 0.3848183 0.04476563 0.10439746  
## 13 0.18353576 0.7292582 0.3848183 0.04476563 0.10439746  
## 14 0.19883041 0.7292582 0.3848183 0.04476563 0.10439746  
## 15 0.21412506 0.7292582 0.3848183 0.04476563 0.10439746  
## 16 0.22941970 0.7095248 0.3181550 0.04400892 0.14406002  
## 17 0.24471435 0.6997532 0.2798752 0.04453018 0.15971209  
## 18 0.26000900 0.6869700 0.2286345 0.04549448 0.17216382  
## 19 0.27530364 0.6764315 0.1830244 0.04417030 0.17574852  
## 20 0.29059829 0.6671238 0.1472563 0.03702976 0.16288159

prp(fit2$finalModel, extra = 106, type = 2)



The resulting tree (while a bit boring looking) is actually quite interesting. This suggests that salinity alone may be the best predictor of island\_side. While relatively simple, this prediction is consistent with the known seperation of water quality data from bayside and oceanside waters and is thus not suprisingly the strongest factor for island\_side.

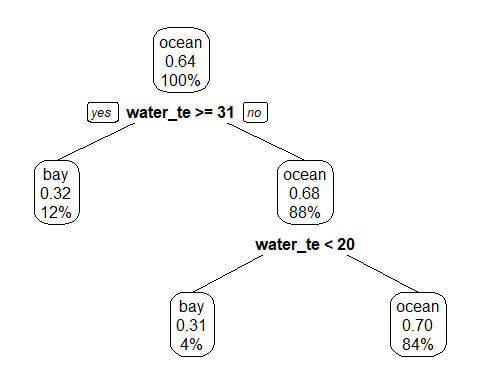
Now that we know salinity is the most important factor for determining island\_side lets delve into the other predictors with salinity removed.

data\_train\_no\_sal <- data\_train %>% dplyr::select(-salinity)  
data\_test\_no\_sal <- data\_test %>% dplyr::select(-salinity)

set.seed(1111)   
outcome = "island\_side"  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
Npred <- ncol(data\_train\_no\_sal)-1  
resultmat <- data.frame(Variable = names(data\_train\_no\_sal)[-1], Accuracy = rep(0,Npred))  
for (n in 2:ncol(data\_train\_no\_sal))  
{  
 fit\_no\_sal <- caret::train( as.formula(paste(outcomename, "~",names(data\_train\_no\_sal)[n])) , data = data\_train, method = "rpart", trControl = fitControl, na.action = na.pass, tuneLength = 20)   
resultmat[n-1,2]= max(fit\_no\_sal$results$Accuracy)   
}  
  
print(resultmat)

## Variable Accuracy  
## 1 Month 0.6400157  
## 2 Day 0.6312873  
## 3 Year 0.6400184  
## 4 military\_time 0.6819280  
## 5 group\_name 0.6627287  
## 6 ph 0.6184522  
## 7 ammonia 0.6239860  
## 8 dissolved\_oxygen 0.6528715  
## 9 water\_temp 0.6794330

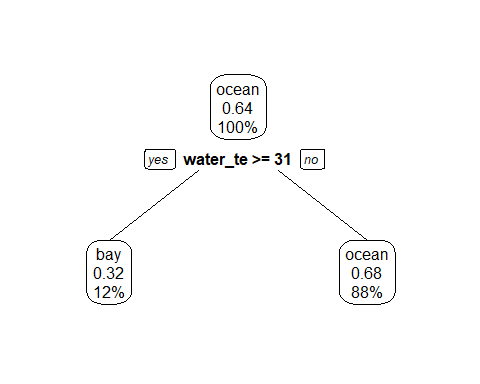
prp(fit\_no\_sal$finalModel, extra = 106, type = 2)

 Interesting, it looks like in the absence of salinity water temperature is the second strongest predictor of island side. Water splits into two distinct groups in this tree. First water teperature greater than or equal to 31 degrees is associated with the bayside. Additionally water less than 20 degrees is also associated with the bayside. This is consistent with known temperature fluctuations and our violin plots which suggests that bayside waters are eurythermal. Lets see if the full fit changes anything.

set.seed(1111)  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
  
fit\_no\_sal\_full = caret::train(island\_side ~ ., data=data\_train\_no\_sal, method="rpart", trControl = fitControl, na.action = na.pass, tuneLength = 20)   
  
print(fit\_no\_sal\_full$results)

## cp Accuracy Kappa AccuracySD KappaSD  
## 1 0.000000000 0.6495146 0.23325292 0.05409807 0.11934246  
## 2 0.006297796 0.6507453 0.23016783 0.05418607 0.12132345  
## 3 0.012595592 0.6421384 0.19327720 0.04820986 0.11613156  
## 4 0.018893387 0.6500641 0.19751918 0.04573561 0.12499661  
## 5 0.025191183 0.6500822 0.18366281 0.04351304 0.11527771  
## 6 0.031488979 0.6481503 0.16759774 0.03558548 0.09837327  
## 7 0.037786775 0.6449974 0.14052141 0.03336579 0.09408125  
## 8 0.044084570 0.6505178 0.13674421 0.03648058 0.09523760  
## 9 0.050382366 0.6523639 0.13930732 0.03578984 0.09497746  
## 10 0.056680162 0.6529796 0.13659191 0.03896377 0.10454186  
## 11 0.062977958 0.6529796 0.13659191 0.03896377 0.10454186  
## 12 0.069275753 0.6542107 0.12207689 0.03818430 0.11186608  
## 13 0.075573549 0.6487002 0.09864991 0.02877343 0.09494994  
## 14 0.081871345 0.6487002 0.09864991 0.02877343 0.09494994  
## 15 0.088169141 0.6462002 0.08809301 0.02647069 0.09037015  
## 16 0.094466937 0.6462002 0.08809301 0.02647069 0.09037015  
## 17 0.100764732 0.6449881 0.08101818 0.02618674 0.09005320  
## 18 0.107062528 0.6449881 0.08101818 0.02618674 0.09005320  
## 19 0.113360324 0.6425638 0.07095015 0.02387751 0.08404543  
## 20 0.119658120 0.6382080 0.04652746 0.01954735 0.06980700

prp(fit\_no\_sal\_full$finalModel, extra = 106, type = 2)



The full fit drops the second temperature branch, this is likely due to a low number of observations below 20 degrees, which is extremely cold for Key Largo however not unreasonable.

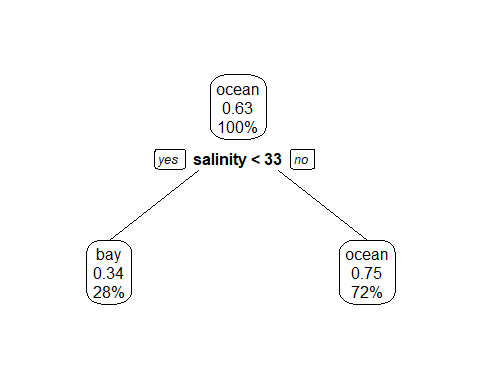
## Model Testing

Lastly, lets evaluate our test data with the same models and generate confusion matricies.

set.seed(1111)   
outcome = "island\_side"  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
Npred <- ncol(data\_test)-1  
resultmat <- data.frame(Variable = names(data\_test)[-1], Accuracy = rep(0,Npred))  
for (n in 2:ncol(data\_test))  
{  
 fit\_test\_single <- caret::train( as.formula(paste(outcomename, "~",names(data\_test)[n])) , data = data\_test, method = "rpart", trControl = fitControl, na.action = na.pass, tuneLength = 20)   
resultmat[n-1,2]= max(fit\_test\_single$results$Accuracy)   
}  
  
print(resultmat)

## Variable Accuracy  
## 1 Month 0.6343020  
## 2 Day 0.5996581  
## 3 Year 0.6270655  
## 4 military\_time 0.6808547  
## 5 group\_name 0.6344729  
## 6 ph 0.6104274  
## 7 ammonia 0.6255840  
## 8 dissolved\_oxygen 0.7087179  
## 9 water\_temp 0.6433048  
## 10 salinity 0.7149288

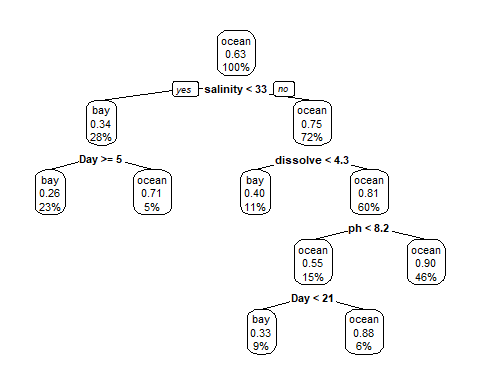
prp(fit\_test\_single$finalModel, extra = 106, type = 2)

 Single predictor assessment is pretty similar, however it seperated the data at 33 ppt salinity.

set.seed(1111)  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
  
fit\_test\_multi = caret::train(island\_side ~ ., data=data\_test, method="rpart", trControl = fitControl, na.action = na.pass, tuneLength = 20)   
  
print(fit\_test\_multi$results)

## cp Accuracy Kappa AccuracySD KappaSD  
## 1 0.00000000 0.6390313 0.2247933 0.07685380 0.1472998  
## 2 0.01288937 0.6405128 0.2263065 0.07776034 0.1488473  
## 3 0.02577873 0.6538462 0.2405334 0.07866314 0.1533283  
## 4 0.03866810 0.6612536 0.2419751 0.06578848 0.1426034  
## 5 0.05155747 0.6492877 0.2217743 0.07479537 0.1493138  
## 6 0.06444683 0.6478063 0.2178347 0.07480360 0.1492802  
## 7 0.07733620 0.6597151 0.2463113 0.07369451 0.1370706  
## 8 0.09022556 0.6552707 0.2345517 0.07272402 0.1332395  
## 9 0.10311493 0.6537322 0.2241980 0.07065168 0.1293978  
## 10 0.11600430 0.6521937 0.2122590 0.07278059 0.1538368  
## 11 0.12889366 0.6521937 0.2122590 0.07278059 0.1538368  
## 12 0.14178303 0.6521937 0.2122590 0.07278059 0.1538368  
## 13 0.15467240 0.6521937 0.2122590 0.07278059 0.1538368  
## 14 0.16756176 0.6521937 0.2122590 0.07278059 0.1538368  
## 15 0.18045113 0.6536752 0.2079347 0.07188832 0.1583400  
## 16 0.19334049 0.6536752 0.2079347 0.07188832 0.1583400  
## 17 0.20622986 0.6447863 0.1743151 0.06624044 0.1522146  
## 18 0.21911923 0.6447863 0.1743151 0.06624044 0.1522146  
## 19 0.23200859 0.6402849 0.1257387 0.05680189 0.1512656  
## 20 0.24489796 0.6402849 0.1257387 0.05680189 0.1512656

prp(fit\_test\_multi$finalModel, extra = 106, type = 2)



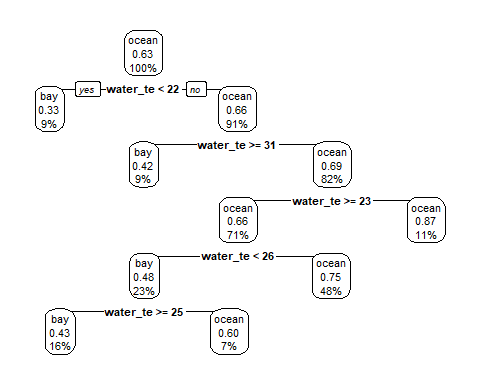
Multi predictor seperates at 33 ppt as well with a small change in the percentages. Regardless we can confidantly state that salinity is the strongest measure of water quality to determine island side.

And now to remove salinity.

set.seed(1111)   
outcome = "island\_side"  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
Npred <- ncol(data\_test\_no\_sal)-1  
resultmat <- data.frame(Variable = names(data\_test\_no\_sal)[-1], Accuracy = rep(0,Npred))  
for (n in 2:ncol(data\_test\_no\_sal))  
{  
 fit\_test\_single\_no\_sal <- caret::train( as.formula(paste(outcomename, "~",names(data\_test\_no\_sal)[n])) , data = data\_test\_no\_sal, method = "rpart", trControl = fitControl, na.action = na.pass, tuneLength = 20)   
resultmat[n-1,2]= max(fit\_test\_single\_no\_sal$results$Accuracy)   
}  
  
print(resultmat)

## Variable Accuracy  
## 1 Month 0.6343020  
## 2 Day 0.5996581  
## 3 Year 0.6270655  
## 4 military\_time 0.6808547  
## 5 group\_name 0.6344729  
## 6 ph 0.6104274  
## 7 ammonia 0.6255840  
## 8 dissolved\_oxygen 0.7087179  
## 9 water\_temp 0.6433048

prp(fit\_test\_single\_no\_sal$finalModel, extra = 106, type = 2)

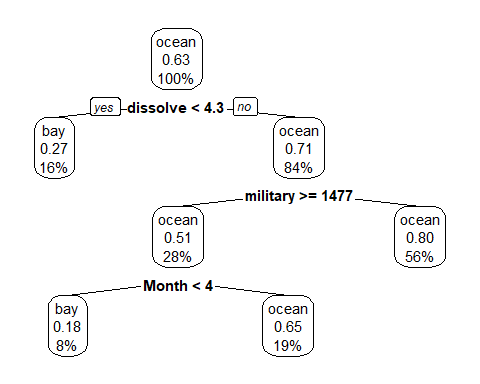


Test data seperates water only by the lowest water temperature at 22 degrees.

set.seed(1111)  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
  
fit\_test\_multi\_no\_sal = caret::train(island\_side ~ ., data=data\_test\_no\_sal, method="rpart", trControl = fitControl, na.action = na.pass, tuneLength = 20)   
  
print(fit\_test\_multi\_no\_sal$results)

## cp Accuracy Kappa AccuracySD KappaSD  
## 1 0.00000000 0.6197151 0.18147643 0.08790405 0.1637622  
## 2 0.01074114 0.6256410 0.18844621 0.08576536 0.1622550  
## 3 0.02148228 0.6331054 0.20024096 0.09418429 0.1820676  
## 4 0.03222342 0.6419943 0.19849360 0.07952019 0.1613458  
## 5 0.04296455 0.6419943 0.19271496 0.07806944 0.1583979  
## 6 0.05370569 0.6391453 0.17010317 0.08646415 0.1688230  
## 7 0.06444683 0.6332194 0.14314233 0.08425971 0.1772710  
## 8 0.07518797 0.6347578 0.14451627 0.08572197 0.1788273  
## 9 0.08592911 0.6332764 0.13862510 0.07922017 0.1636547  
## 10 0.09667025 0.6317949 0.13593858 0.07963642 0.1639998  
## 11 0.10741139 0.6392023 0.14580533 0.08014273 0.1673149  
## 12 0.11815252 0.6377208 0.14089559 0.07995599 0.1698023  
## 13 0.12889366 0.6377208 0.13266029 0.07995599 0.1715032  
## 14 0.13963480 0.6377208 0.13266029 0.07995599 0.1715032  
## 15 0.15037594 0.6377208 0.13266029 0.07995599 0.1715032  
## 16 0.16111708 0.6377208 0.12727568 0.07995599 0.1735404  
## 17 0.17185822 0.6377208 0.12727568 0.07995599 0.1735404  
## 18 0.18259936 0.6345869 0.10136420 0.06754265 0.1496577  
## 19 0.19334049 0.6345869 0.10136420 0.06754265 0.1496577  
## 20 0.20408163 0.6284330 0.07160934 0.06150245 0.1273014

prp(fit\_test\_multi\_no\_sal$finalModel, extra = 106, type = 2)

 Humm interesting in the absence of salinity, dissolved oxygen is now the dominat predictor in the test data.

All very interesting analyses, but after all of that it looks like a simple measure of salinity (fit2) is the best method to determine the isalnd side of a sample. This is consistent with our prior data visualizations and with known characteristics of esturarine and marine waters.

Lets try a prediction.

train\_predict <- predict(fit2, data\_train)  
confusionMatrix(train\_predict, data\_train$island\_side)$table

## Reference  
## Prediction bay ocean  
## bay 60 26  
## ocean 57 182

test\_predict <- predict(fit2, data\_test)  
confusionMatrix(test\_predict, data\_test$island\_side)$table

## Reference  
## Prediction bay ocean  
## bay 18 10  
## ocean 31 75

It looks like our model is reasonably effective at identifying ocean side sites, however a bit worse when trying to classify bayside sites.

Perhaps a predictor of site type may be more appropriate for this study.

glimpse(model\_ready)

## Observations: 513  
## Variables: 13  
## $ island\_side <fct> ocean, ocean, ocean, ocean, bay, bay, bay, oc...  
## $ Month <dbl> 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ...  
## $ Day <dbl> 8, 8, 8, 8, 9, 9, 12, 17, 27, 27, 27, 27, 27,...  
## $ Year <dbl> 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 1...  
## $ military\_time <dbl> 1415, 1515, 1550, 1555, 1015, 1022, 1035, 100...  
## $ location <fct> Boat Ramp, Grecian Dry Rocks, Grecian Dry Roc...  
## $ group\_name <fct> NA, McLean High School, McLean High School, M...  
## $ ph <dbl> 8.0, 8.4, 8.2, 8.4, 8.0, 8.0, 8.0, 8.4, 8.2, ...  
## $ ammonia <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0...  
## $ dissolved\_oxygen <dbl> 5.0, 4.0, 4.0, 6.0, 4.0, 5.0, 6.0, 6.0, 5.0, ...  
## $ water\_temp <dbl> NA, 23.5, 21.0, 36.0, 18.0, 18.0, 18.3, 23.0,...  
## $ salinity <dbl> 36, 40, 44, 35, 30, 33, 35, 40, 30, 35, 25, 3...  
## $ site\_type <fct> Seagrass/Mangrove, Coral Reef, Coral Reef, Co...

outcome\_2 <- model\_ready %>% dplyr::select(-location, -island\_side)  
  
outcome\_2 <- na.omit(outcome\_2)

outcome\_2 <- outcome\_2[, c(11,1,2,3,4,5,6,7,8,9,10)]

set.seed(123)  
  
trainset <- caret::createDataPartition(y = outcome\_2$site\_type, p = 0.7, list = FALSE)  
  
#Extract observations/rows for training and assign to new variable  
data\_train\_2 = outcome\_2[trainset,]   
  
#Same as above but for the test set  
data\_test\_2 = outcome\_2[-trainset,]

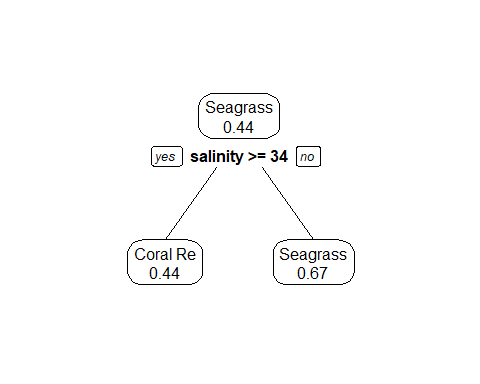
library(mlr) #We will jump between mlr and caret so call with :: at all times.   
  
mlr::measureACC("Seagrass/Mangrove", outcome\_2$site\_type)

## [1] 0.4400871

set.seed(1111) #To keep reproducibility  
  
outcomename = "site\_type" #Name of outcome of interest  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5) #Sets K-fold cross-validation.  
  
Npred <- ncol(data\_train)-1 #Sets predictors minus the outcome in column 1.  
  
resultmat <- data.frame(Variable = names(data\_train\_2)[-1], Accuracy = rep(0,Npred)) #Takes results and moves to dataframe named resultmat, measure set to accuracy.  
  
for (n in 2:ncol(data\_train\_2))  
{  
 fit11 <- caret::train( as.formula(paste(outcomename, "~",names(data\_train\_2)[n])) , data = data\_train\_2, method = "rpart", trControl = fitControl, na.action = na.pass, tuneLength = 10)   
resultmat[n-1,2]= max(fit11$results$Accuracy)   
} #Loop to all variables through prediction. Method rpart = tree algorithm, and skipping any NA values.   
  
print(resultmat)

## Variable Accuracy  
## 1 Month 0.4413879  
## 2 Day 0.4786054  
## 3 Year 0.4277451  
## 4 military\_time 0.4510170  
## 5 group\_name 0.4377851  
## 6 ph 0.5019332  
## 7 ammonia 0.4315565  
## 8 dissolved\_oxygen 0.4403073  
## 9 water\_temp 0.4966244  
## 10 salinity 0.5230600

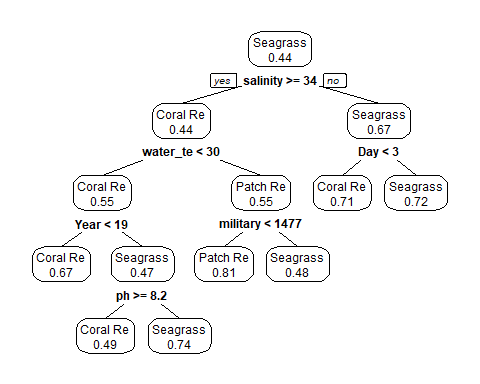
prp(fit11$finalModel, extra = 8, type = 2)



set.seed(1111)  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
  
fit111 = caret::train(site\_type ~ ., data=data\_train\_2, method="rpart", trControl = fitControl, na.action = na.pass, tuneLength = 10)   
  
print(fit111$results)

## cp Accuracy Kappa AccuracySD KappaSD  
## 1 0.00000000 0.5785253 0.32061462 0.04259926 0.06460264  
## 2 0.01718846 0.5900742 0.33337305 0.04829175 0.07850694  
## 3 0.03437692 0.5771496 0.32287221 0.03556029 0.05506861  
## 4 0.05156538 0.5666677 0.31221379 0.03828813 0.05842826  
## 5 0.06875384 0.5573299 0.29596419 0.04319194 0.07709967  
## 6 0.08594230 0.5504038 0.28079821 0.05385597 0.09987635  
## 7 0.10313076 0.5491731 0.27841859 0.05375929 0.09760818  
## 8 0.12031921 0.5127377 0.20831002 0.05614535 0.10650042  
## 9 0.13750767 0.4886614 0.14254839 0.04414447 0.10079818  
## 10 0.15469613 0.4638441 0.08065302 0.03533096 0.08732622

prp(fit111$finalModel, extra = 8, type = 2)

 Swapping outcomes it appears that salinity remains the intial deciding variable. Data is either grouped into Seagrass/Mangrove or not. This grouping makes sense because all bayside sites are Seagrass/Mangrove, however oceanside can be a mix of all three. Coral Reef and Patch Reef/Hardbottom seperates by water temperature primairly which is a reasonable branch as patch reefs tend to be cooler in temperature than coral reefs. Beyond these branches there are a few branches based off of date/year data. While this clustering is interesting to note, it does not clarify much about our overall goal of measuring the efficacy of citizen science data as sampling time/date is not under the control of the citizen scientists. This analysis and the trees above suggest, it would be most prudent to build our trees entirely from the variables that the citizen scientists can control as it would not be appropriate to measure their efficacy with items outside of their control.

## Final Model Fitting Evaluation

We will measure both outcomes island\_side and site\_type with datasets which only contain variables controlled by the citizen scientists.

cs\_only\_side <- model\_ready %>% dplyr::select(-Month, -Day, -Year, -military\_time, -location, -group\_name, -site\_type)  
  
cs\_only\_side <- na.omit(cs\_only\_side)  
  
glimpse(cs\_only\_side)

## Observations: 475  
## Variables: 6  
## $ island\_side <fct> ocean, ocean, ocean, bay, bay, bay, ocean, ba...  
## $ ph <dbl> 8.4, 8.2, 8.4, 8.0, 8.0, 8.0, 8.4, 8.2, 8.0, ...  
## $ ammonia <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0...  
## $ dissolved\_oxygen <dbl> 4.0, 4.0, 6.0, 4.0, 5.0, 6.0, 6.0, 5.0, 5.0, ...  
## $ water\_temp <dbl> 23.5, 21.0, 36.0, 18.0, 18.0, 18.3, 23.0, 18....  
## $ salinity <dbl> 40, 44, 35, 30, 33, 35, 40, 30, 35, 25, 35, 3...

set.seed(123)  
  
trainset <- caret::createDataPartition(y = cs\_only\_side$island\_side, p = 0.7, list = FALSE)  
  
#Extract observations/rows for training and assign to new variable  
data\_train\_cs\_1 = cs\_only\_side[trainset,]   
  
#Same as above but for the test set  
data\_test\_cs\_2 = cs\_only\_side[-trainset,]

mlr::measureACC("ocean", cs\_only\_side$island\_side)

## [1] 0.6336842

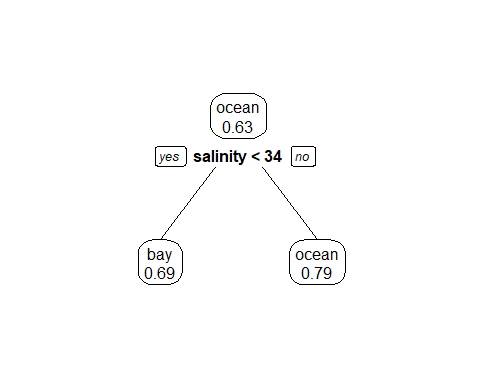
set.seed(1111) #To keep reproducibility  
  
outcomename = "island\_side" #Name of outcome of interest  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5) #Sets K-fold cross-validation.  
  
Npred <- ncol(data\_train\_cs\_1)-1 #Sets predictors minus the outcome in column 1.  
  
resultmat <- data.frame(Variable = names(data\_train\_cs\_1)[-1], Accuracy = rep(0,Npred)) #Takes results and moves to dataframe named resultmat, measure set to accuracy.  
  
for (n in 2:ncol(data\_train\_cs\_1))  
{  
 fit22 <- caret::train( as.formula(paste(outcomename, "~",names(data\_train\_cs\_1)[n])) , data = data\_train\_cs\_1, method = "rpart", trControl = fitControl, na.action = na.pass, tuneLength = 10)   
resultmat[n-1,2]= max(fit22$results$Accuracy)   
} #Loop to all variables through prediction. Method rpart = tree algorithm, and skipping any NA values.   
  
print(resultmat)

## Variable Accuracy  
## 1 ph 0.6053484  
## 2 ammonia 0.6276665  
## 3 dissolved\_oxygen 0.6517102  
## 4 water\_temp 0.6877753  
## 5 salinity 0.7512943

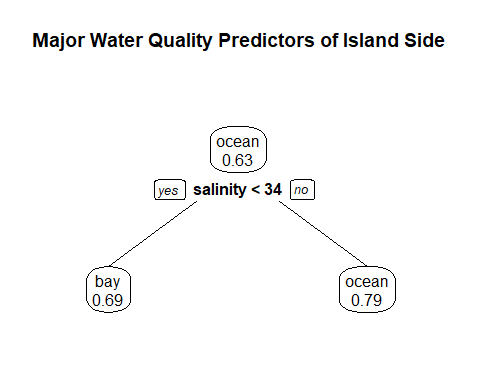
set.seed(1111)  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
  
fit222 = caret::train(island\_side ~ ., data=data\_train\_cs\_1, method="rpart", trControl = fitControl, na.action = na.pass, tuneLength = 10)   
  
print(fit222$results)

## cp Accuracy Kappa AccuracySD KappaSD  
## 1 0.00000000 0.7117102 0.3623902 0.04981565 0.1124531  
## 2 0.03825137 0.7448360 0.4378140 0.04979330 0.1093395  
## 3 0.07650273 0.7502724 0.4465044 0.04996657 0.1118168  
## 4 0.11475410 0.7502724 0.4465044 0.04996657 0.1118168  
## 5 0.15300546 0.7502724 0.4465044 0.04996657 0.1118168  
## 6 0.19125683 0.7502724 0.4465044 0.04996657 0.1118168  
## 7 0.22950820 0.7502724 0.4465044 0.04996657 0.1118168  
## 8 0.26775956 0.7502724 0.4465044 0.04996657 0.1118168  
## 9 0.30601093 0.7417876 0.4195996 0.05060036 0.1338864  
## 10 0.34426230 0.6798577 0.2051851 0.04365830 0.1923397

prp(fit222$finalModel, extra = 8, type = 2)



cs\_not\_best\_tree <- prp(fit222$finalModel, extra = 8, type = 2, main = "Major Water Quality Predictors of Island Side")



ww=17.8/2.54; wh=ww; #for saving plot  
  
dev.print(device=png,width=ww,height=wh,units="in",res=600,file="cs\_not\_best\_tree.png") #save tree to file

## png   
## 2

file.move("./cs\_not\_best\_tree.png", "../../results")

## 0 files moved. 1 failed.

## Some files failed to move because it would have caused files to be overwritten.   
## \* To allow overwriting, use `overwrite = TRUE`.

Well it appears that salinity prevails once again! The overwhelming success of salinity as a predictor suggests that the issue here is not with our predictors but actually with our outcome. Island side is too strongly influenced by salinity that our model cannot develop a method by which better accuracy is produced through the addition of further variables. This is very interesting information, however it does not address our overall objective to measure the efficacy of citizen science data collection. For this assessment, our alternate outcome of site type is more appropriate.

We will end with a model evauation of site type.

cs\_only\_site <- outcome\_2 %>% dplyr::select(-Month, -Day, -Year, -military\_time, -group\_name)  
  
cs\_only\_side <- na.omit(cs\_only\_site)  
  
glimpse(cs\_only\_site)

## Observations: 459  
## Variables: 6  
## $ site\_type <fct> Coral Reef, Coral Reef, Coral Reef, Seagrass/...  
## $ ph <dbl> 8.4, 8.2, 8.4, 8.0, 8.0, 8.0, 8.4, 8.2, 8.0, ...  
## $ ammonia <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...  
## $ dissolved\_oxygen <dbl> 4.0, 4.0, 6.0, 4.0, 5.0, 6.0, 6.0, 5.0, 5.0, ...  
## $ water\_temp <dbl> 23.5, 21.0, 36.0, 18.0, 18.0, 18.3, 23.0, 18....  
## $ salinity <dbl> 40, 44, 35, 30, 33, 35, 40, 30, 35, 25, 35, 3...

set.seed(123)  
  
trainset <- caret::createDataPartition(y = cs\_only\_site$site\_type, p = 0.7, list = FALSE)  
  
#Extract observations/rows for training and assign to new variable  
data\_train\_cs\_2 = cs\_only\_site[trainset,]   
  
#Same as above but for the test set  
data\_test\_cs\_2 = cs\_only\_site[-trainset,]

mlr::measureACC("Seagrass/Mangrove", cs\_only\_site$site\_type)

## [1] 0.4400871

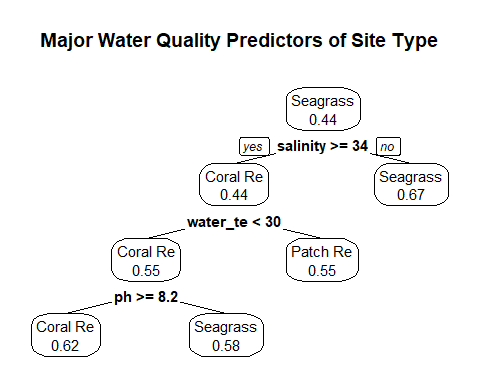
set.seed(1111) #To keep reproducibility  
  
outcomename = "site\_type" #Name of outcome of interest  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5) #Sets K-fold cross-validation.  
  
Npred <- ncol(data\_train\_cs\_2)-1 #Sets predictors minus the outcome in column 1.  
  
resultmat <- data.frame(Variable = names(data\_train\_cs\_2)[-1], Accuracy = rep(0,Npred)) #Takes results and moves to dataframe named resultmat, measure set to accuracy.  
  
for (n in 2:ncol(data\_train\_cs\_2))  
{  
 fit33 <- caret::train( as.formula(paste(outcomename, "~",names(data\_train\_cs\_2)[n])) , data = data\_train\_cs\_2, method = "rpart", trControl = fitControl, na.action = na.pass, tuneLength = 10)   
resultmat[n-1,2]= max(fit33$results$Accuracy)   
} #Loop to all variables through prediction. Method rpart = tree algorithm, and skipping any NA values.   
  
print(resultmat)

## Variable Accuracy  
## 1 ph 0.5021142  
## 2 ammonia 0.4285013  
## 3 dissolved\_oxygen 0.4562355  
## 4 water\_temp 0.5026544  
## 5 salinity 0.5232880

set.seed(1111)  
  
fitControl <- trainControl(method="repeatedcv",number=5,repeats=5)   
  
fit333 = caret::train(site\_type ~ ., data=data\_train\_cs\_2, method="rpart", trControl = fitControl, na.action = na.pass, tuneLength = 10)   
  
print(fit333$results)

## cp Accuracy Kappa AccuracySD KappaSD  
## 1 0.00000000 0.5678317 0.31352926 0.05042386 0.07904404  
## 2 0.01718846 0.5887743 0.33613092 0.03847198 0.06119446  
## 3 0.03437692 0.5956479 0.35199178 0.04652066 0.07137589  
## 4 0.05156538 0.5653588 0.31013682 0.03623826 0.05595739  
## 5 0.06875384 0.5573299 0.29596419 0.04319194 0.07709967  
## 6 0.08594230 0.5504038 0.28079821 0.05385597 0.09987635  
## 7 0.10313076 0.5491731 0.27841859 0.05375929 0.09760818  
## 8 0.12031921 0.5127377 0.20831002 0.05614535 0.10650042  
## 9 0.13750767 0.4886614 0.14254839 0.04414447 0.10079818  
## 10 0.15469613 0.4638441 0.08065302 0.03533096 0.08732622

cs\_best\_tree <- prp(fit333$finalModel, extra = 8, type = 2, main = "Major Water Quality Predictors of Site Type")



cs\_best\_tree

## $obj  
## n= 323   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 323 181 Seagrass/Mangrove (0.36842105 0.19195046 0.43962848)   
## 2) salinity>=33.75 204 114 Coral Reef (0.44117647 0.25490196 0.30392157)   
## 4) water\_temp< 29.55 153 69 Coral Reef (0.54901961 0.15686275 0.29411765)   
## 8) ph>=8.15 115 44 Coral Reef (0.61739130 0.18260870 0.20000000) \*  
## 9) ph< 8.15 38 16 Seagrass/Mangrove (0.34210526 0.07894737 0.57894737) \*  
## 5) water\_temp>=29.55 51 23 Patch Reef/Hardbottom (0.11764706 0.54901961 0.33333333) \*  
## 3) salinity< 33.75 119 39 Seagrass/Mangrove (0.24369748 0.08403361 0.67226891) \*  
##   
## $snipped.nodes  
## NULL  
##   
## $xlim  
## [1] 0 1  
##   
## $ylim  
## [1] 0 1  
##   
## $x  
## [1] 0.70311491 0.48565834 0.27016084 0.07425402 0.46606766 0.70115584  
## [7] 0.92057148  
##   
## $y  
## [1] 0.90252862 0.61806790 0.33360717 0.04914645 0.04914645 0.33360717  
## [7] 0.61806790  
##   
## $branch.x  
## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## x 0.7031149 0.4856583 0.2701608 0.07425402 0.4660677 0.7011558 0.9205715  
## NA 0.6596236 0.4425588 0.23097948 0.3093422 0.5287578 0.7466062  
## NA 0.7031149 0.4856583 0.27016084 0.2701608 0.4856583 0.7031149  
##   
## $branch.y  
## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## y 1.005421 0.7209601 0.4364994 0.1520387 0.1520387 0.4364994 0.7209601  
## NA 0.7847433 0.5002826 0.2158219 0.2158219 0.5002826 0.7847433  
## NA 0.7847433 0.5002826 0.2158219 0.2158219 0.5002826 0.7847433  
##   
## $labs  
## [1] "Seagrass\n0.44" "Coral Re\n0.44" "Coral Re\n0.55" "Coral Re\n0.62"  
## [5] "Seagrass\n0.58" "Patch Re\n0.55" "Seagrass\n0.67"  
##   
## $cex  
## [1] 0.9640625  
##   
## $boxes  
## $boxes$x1  
## [1] 0.61262576 0.40128332 0.18578582 -0.01012099 0.37557851 0.61311234  
## [7] 0.83008233  
##   
## $boxes$y1  
## [1] 0.844938412 0.560477691 0.276016970 -0.008443751 -0.008443751  
## [6] 0.276016970 0.560477691  
##   
## $boxes$x2  
## [1] 0.7936041 0.5700334 0.3545359 0.1586290 0.5565568 0.7891993 1.0110606  
##   
## $boxes$y2  
## [1] 1.0054209 0.7209601 0.4364994 0.1520387 0.1520387 0.4364994 0.7209601  
##   
##   
## $split.labs  
## [1] ""  
##   
## $split.cex  
## [1] 1 1 1 1 1 1 1  
##   
## $split.box  
## $split.box$x1  
## [1] 0.5759410 0.3548159 0.1760032 NA NA NA NA  
##   
## $split.box$y1  
## [1] 0.7394413 0.4549806 0.1705199 NA NA NA NA  
##   
## $split.box$x2  
## [1] 0.8302889 0.6165008 0.3643185 NA NA NA NA  
##   
## $split.box$y2  
## [1] 0.8300454 0.5455846 0.2611239 NA NA NA NA

ww=17.8/2.54; wh=ww; #for saving plot  
  
dev.print(device=png,width=ww,height=wh,units="in",res=600,file="cs\_best\_tree.png") #save tree to file

## png   
## 2

file.move("./cs\_best\_tree.png", "../../results")

## 0 files moved. 1 failed.

## Some files failed to move because it would have caused files to be overwritten.   
## \* To allow overwriting, use `overwrite = TRUE`.

Much better! Based on this tree it looks like we have identified the model that characterizes our outcome by a cascade of our water quality predictors. Consistent with our models above, salinity remains the primary branch seperating low salinity into Seagrass/Mangrove (likely all of our bayside sites) and high salinity into a mixture of all three site types. The second branch divdes along temperature lines which is once again consistent with the expected divisions as patch reefs tend to be cooler sites. The final split divides high saline and high temperature sites into coral reef and seagrass respectively. This division is based upon pH measures with greater pH representing Coral Reefs and lower pH representing Seagrass/Mangrove. This split is likely due to the average depth and proximity of Seagrass/Mangroves to shore and thus more strongly influenced by runoff/non point source pollution/ and rainfall. Though the accuracy of this model is similar to the accuracy of the model before removing all non-citizen science determined variables it is important to note that this model better represents the efficacy of citizen science data by eliminating confounding influences of data splitting.

Finally we will evaluate our test data with our model to produce a confusion matrix.

train\_predict <- predict(fit333, data\_train\_cs\_2)  
confusionMatrix(train\_predict, data\_train\_cs\_2$site\_type)$table

## Reference  
## Prediction Coral Reef Patch Reef/Hardbottom Seagrass/Mangrove  
## Coral Reef 71 21 23  
## Patch Reef/Hardbottom 6 28 17  
## Seagrass/Mangrove 42 13 102

test\_predict <- predict(fit333, data\_test\_cs\_2)  
confusionMatrix(test\_predict, data\_test\_cs\_2$site\_type)$table

## Reference  
## Prediction Coral Reef Patch Reef/Hardbottom Seagrass/Mangrove  
## Coral Reef 31 13 17  
## Patch Reef/Hardbottom 9 5 11  
## Seagrass/Mangrove 10 8 32

Our model is not a very efficent at predicting the correct site type, however this is okay since the overall goal was not to make predictions but rather determine the major splitting points in the data. These major splitting points were serve as our measures of citizen science data collection, if they are consistent with the factors that typically classify our three site types of interest this will suggest that the data collected is indeed accurate.