Unsupervised Learning Analysis

William Norfolk

11/18/2019

This script expands on the findings of seasonal exploratory and bivariate analysis through the usage of tree based modeling to express seasonality.

Start by loading required libraries

library(readxl)  
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(tidyverse)

## -- Attaching packages ------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1 v readr 1.3.1  
## v tibble 2.1.3 v purrr 0.3.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v ggplot2 3.2.1 v forcats 0.4.0

## -- Conflicts ---------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(forcats)  
library(ggthemes)  
library(knitr)  
library(naniar)  
library(broom)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(ggpubr)

## Loading required package: magrittr

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':  
##   
## set\_names

## The following object is masked from 'package:tidyr':  
##   
## extract

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(mlr)

## Loading required package: ParamHelpers

##   
## Attaching package: 'mlr'

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

library(rpart)  
library(rpart.plot)  
library(ranger)  
library(visdat)  
  
#Note it will be very important to call functions using the :: library notation due to the frequency of function masking by our libraries.

Next load the data a 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...

## Prep/Clean the Data

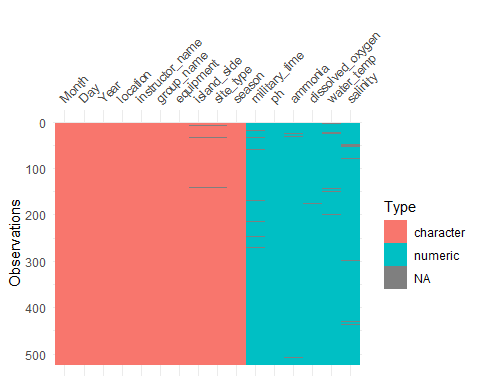
For our previous analyses of season we have grouped the data by month to increase the number of monthly replicates and visualize linear changes in the data by individual months. For seasonality modeling we will bin the months into season categories through the addition of a new variable “season”. Months 12-2 will represent winter, 3-5 will represent spring, 6-8 will represent summer, and 9-11 will represent fall.

Before we can begin moeling we must first create a new season variable.

add\_season <- WQ\_clean\_data %>% dplyr::mutate(season = recode(Month,  
 "12" = "Winter",  
 "01" = "Winter",  
 "02" = "Winter",  
 "03" = "Spring",  
 "04" = "Spring",  
 "05" = "Spring",  
 "06" = "Summer",  
 "07" = "Summer",  
 "08" = "Summer",  
 "09" = "Fall",  
 "10" = "Fall",  
 "11" = "Fall"))

Next we lets check our data to ensure everything looks ok before we begin modeling.

vis\_dat(add\_season)



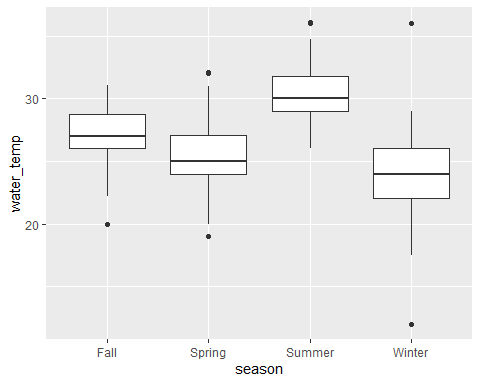
A few NAs are present in the data but no strong areas of NAs are indicated. Tree models should be able to handle baseline amounts of NA values so for now we will leave them be. As for our variable classes all of our water quality parameters are numeric as they should be so no modification is needed there. Season is our outcome in this case so we must change this to a factor. All remaining variables are character class, however we will remove these before modeling since they are not water quality determinants of season.

add\_season$season <- as.factor(as.character(add\_season$season))

Next lets take a look at some visualizations of the new season variable before we begin modeling to see if there is any further cleaning that is needed.

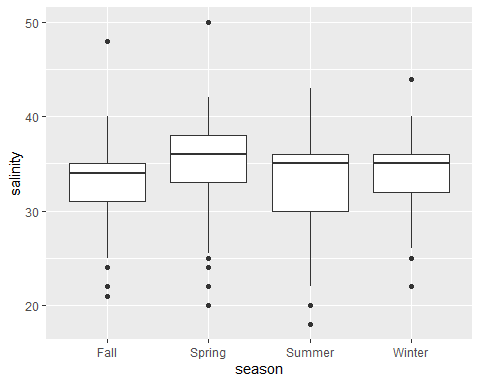
season\_temp <- ggplot(add\_season, aes(x = season, y = water\_temp)) + geom\_boxplot()  
  
season\_temp

## Warning: Removed 12 rows containing non-finite values (stat\_boxplot).



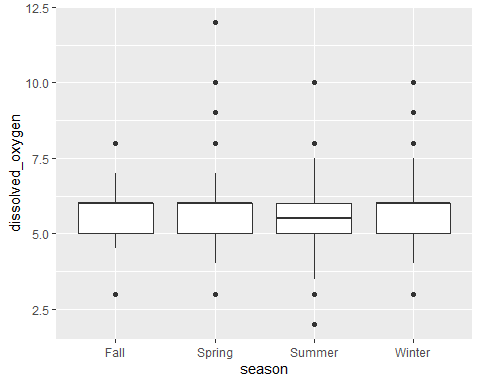
season\_sal <- ggplot(add\_season, aes(x = season, y = salinity)) + geom\_boxplot()  
  
season\_sal

## Warning: Removed 16 rows containing non-finite values (stat\_boxplot).



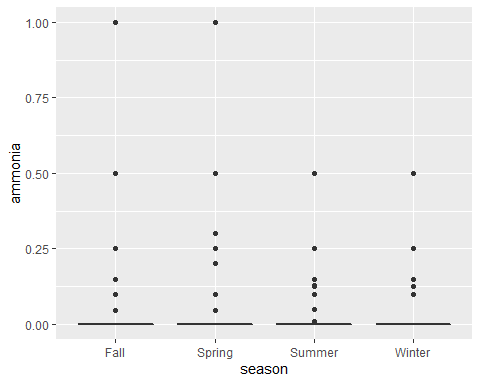
season\_do <- ggplot(add\_season, aes(x = season, y = dissolved\_oxygen)) + geom\_boxplot()  
  
season\_do

## Warning: Removed 4 rows containing non-finite values (stat\_boxplot).



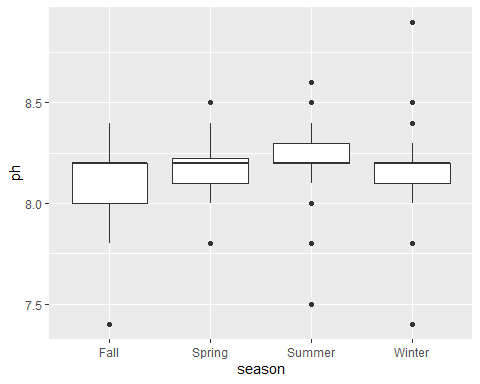
season\_amm <- ggplot(add\_season, aes(x = season, y = ammonia)) + geom\_boxplot()  
  
season\_amm

## Warning: Removed 10 rows containing non-finite values (stat\_boxplot).



season\_ph <- ggplot(add\_season, aes(x = season, y = ph)) + geom\_boxplot()  
  
season\_ph

## Warning: Removed 6 rows containing non-finite values (stat\_boxplot).



Humm I am concerned the tree will only build using temperature which is not terribly informative…but we press on

Since our data looks good we are just about ready to model. First we need to remove any variables that are not of interest for seasonal prediction, and move our outcome to the front of the variable list.

reduce\_vars <- add\_season %>% dplyr::select(season, water\_temp, salinity, ammonia, dissolved\_oxygen, ph)  
  
#This code also moves our outcome to the first slot so no additional shuffling code is needed.

Lastly we rename our data for model\_ready to reduce confusion.

model\_ready <- reduce\_vars

## Split the Data

First we will split our data into train and test sets.

set.seed(123)  
  
trainset <- caret::createDataPartition(y = model\_ready$season, 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,]

The split should allocate 367 observations into train and 155 into test.

Next we compute a null model for comparison.

library(mlr) #We will jump between mlr and caret so call with :: at all times.   
  
mlr::measureACC("Summer", model\_ready$season)

## [1] 0.3888889

Next we move into single predictor tree fitting.

set.seed(1111) #To keep reproducibility  
  
outcomename = "season" #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)

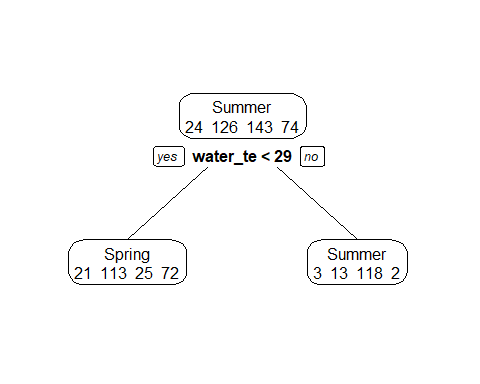
## Variable Accuracy  
## 1 water\_temp 0.6244873  
## 2 salinity 0.4310667  
## 3 ammonia 0.4060200  
## 4 dissolved\_oxygen 0.3825780  
## 5 ph 0.3317797

Not unexpected temperature is the strongest influencer of season.

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

## cp Accuracy Kappa AccuracySD KappaSD  
## 1 0.00000000 0.5887361 0.3827672 0.05269395 0.06789939  
## 2 0.04365079 0.6234062 0.4100063 0.03575859 0.05582165  
## 3 0.08730159 0.6234062 0.4100063 0.03575859 0.05582165  
## 4 0.13095238 0.6234062 0.4100063 0.03575859 0.05582165  
## 5 0.17460317 0.6234062 0.4100063 0.03575859 0.05582165  
## 6 0.21825397 0.6234062 0.4100063 0.03575859 0.05582165  
## 7 0.26190476 0.6234062 0.4100063 0.03575859 0.05582165  
## 8 0.30555556 0.6234062 0.4100063 0.03575859 0.05582165  
## 9 0.34920635 0.6234062 0.4100063 0.03575859 0.05582165  
## 10 0.39285714 0.5063645 0.2074748 0.10671216 0.18940247

prp(fit1$finalModel, extra = 1, type = 2)

 So it looks like water temperature along at 29 degrees is the best fit for determining seasonality.

## Unsupervised Cluster Analysis

Perhaps we are approaching our models from the wrong angle. Since our data is not specifically targeted at particular outcome (i.e. occurance of disease, algal bloom, etc.) and focuses more as a monitoring program; it may be best to use unsupervised modeling techniques to to cluster our data. We will use agglomerative clustering to cluster our data by our specific categorical variables to determine clustering characteristics.

First we must remove any NAs from our data as cluster functions cannot handle them. Additionally, we need to select our branch outcome (in this case location) as well as our five numeric water quality parameters of interest.

cluster\_trim <- WQ\_clean\_data %>% dplyr::select(location, water\_temp, salinity, dissolved\_oxygen, ammonia, ph)  
  
cluster\_trim <- na.omit(cluster\_trim)

We will cluster by site means to produce a readable dendrogram. With a retrospective view, there are three locations which will greatly skew our data when producing a dendrogram. All three locations have either one or few (The Elbow) measures comprising the site. When allowed to remain in our sample set these data points force our dendrogram to subdivide further grouping themselves into indivual clusters. Though this information is interesting, considering the n of each of these locations and the large impact they play on the final dendrogram, we will remove them here to produce a generalized view of the clusters.

location\_average <- aggregate(. ~ location, cluster\_trim, mean)  
  
drop\_buttonwood <- location\_average[!location\_average$location == "Buttonwood", ]  
drop\_orc <- drop\_buttonwood[!drop\_buttonwood$location == "Ocean Reef Club", ]  
drop\_elbow <- drop\_orc[!drop\_orc$location == "The Elbow", ]  
drop\_NA\_site <- drop\_elbow[!drop\_elbow$location == "NA", ]  
  
cluster\_ready <- drop\_NA\_site

We will need to add three additional cluster analysis libraries. Then we produce the dendrograms with the code below.

library(factoextra)

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

library(dendextend)

##   
## ---------------------  
## Welcome to dendextend version 1.12.0  
## Type citation('dendextend') for how to cite the package.  
##   
## Type browseVignettes(package = 'dendextend') for the package vignette.  
## The github page is: https://github.com/talgalili/dendextend/  
##   
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues  
## Or contact: <tal.galili@gmail.com>  
##   
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))  
## ---------------------

##   
## Attaching package: 'dendextend'

## The following object is masked from 'package:rpart':  
##   
## prune

## The following object is masked from 'package:ggpubr':  
##   
## rotate

## The following object is masked from 'package:stats':  
##   
## cutree

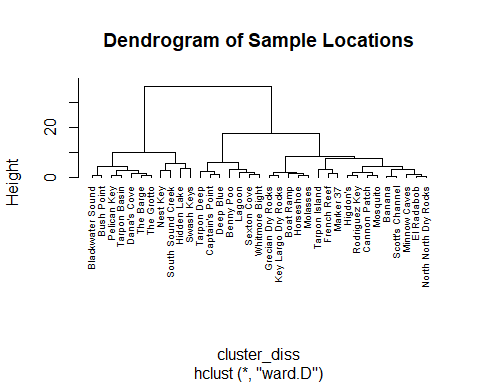
library(cluster)  
  
set.seed(1111) #Ensure reproducible  
  
rownames(cluster\_ready) <- cluster\_ready$location #Assign the location names to rownames to produce dendrograms that show site information.  
  
cluster\_diss <- dist(cluster\_ready, method = "euclidean") #For the hclust function we must first produce a dissimilarity matrix before runing cluster analysis

## Warning in dist(cluster\_ready, method = "euclidean"): NAs introduced by  
## coercion

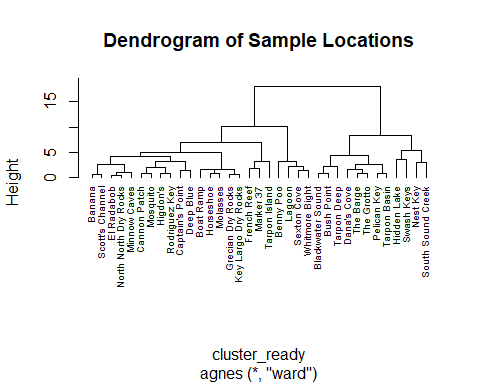
cluster\_site\_hclust <- hclust(cluster\_diss, method = "ward.D") #Hierarchial clustering with Ward's minimum variance  
cluster\_site\_agnes <- agnes(cluster\_ready, method = "ward") #Agglomerative clustering with Ward's minimum variance

## Warning in data.matrix(x): NAs introduced by coercion

#Plot the trees. Note agnes can use pltree or plot but hclust must use plot  
good\_tree\_hclust <- plot(cluster\_site\_hclust, cex = 0.6, hang = -1, main = "Dendrogram of Sample Locations", labels = cluster\_ready$location)



good\_tree\_agnes <- pltree(cluster\_site\_agnes, cex = 0.6, hang = -1, main = "Dendrogram of Sample Locations", labels = cluster\_ready$location)



good\_tree\_hclust

## NULL

good\_tree\_agnes

## NULL

Both of the trees look interesting, but it looks like agglomerative is the best option.

Before moving on we will check our method. We started with Ward’s minimum variance method however there are others we could use (single, complete, and average). We will calculate the agglomerative coefficent for each below. A coefficent closer to 1 indicates a stronger clustering.

cluster\_site\_agnes$ac

## [1] 0.9225503

methods <- c( "average", "single", "complete", "ward")  
names(methods) <- c( "average", "single", "complete", "ward")  
  
coeff <- function(x) {  
 agnes(cluster\_ready, method = x)$ac  
}  
  
map\_dbl(methods, coeff)

## average single complete ward   
## 0.7481487 0.6558795 0.8766297 0.9225503

It looks like Ward’s method is the best option.

Just a quick code bit to count the number of values in each cluster.This is more useful if there is a very high number of branches.

sub\_group\_dendro\_1 <- cutree(cluster\_site\_agnes, k = 2)  
  
table(sub\_group\_dendro\_1)

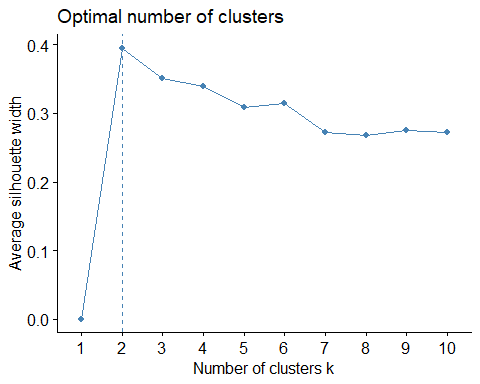
## sub\_group\_dendro\_1  
## 1 2   
## 23 12

Next we will check our optimal number of clusters.

fviz\_nbclust(cluster\_ready, FUN = hcut, method = "silhouette")

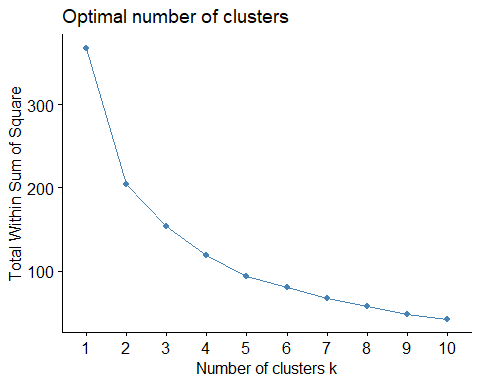
## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion



fviz\_nbclust(cluster\_ready, FUN = hcut, method = "wss")

## Warning in stats::dist(x): NAs introduced by coercion  
  
## Warning in stats::dist(x): NAs introduced by coercion  
  
## Warning in stats::dist(x): NAs introduced by coercion  
  
## Warning in stats::dist(x): NAs introduced by coercion  
  
## Warning in stats::dist(x): NAs introduced by coercion  
  
## Warning in stats::dist(x): NAs introduced by coercion  
  
## Warning in stats::dist(x): NAs introduced by coercion  
  
## Warning in stats::dist(x): NAs introduced by coercion  
  
## Warning in stats::dist(x): NAs introduced by coercion  
  
## Warning in stats::dist(x): NAs introduced by coercion  
  
## Warning in stats::dist(x): NAs introduced by coercion



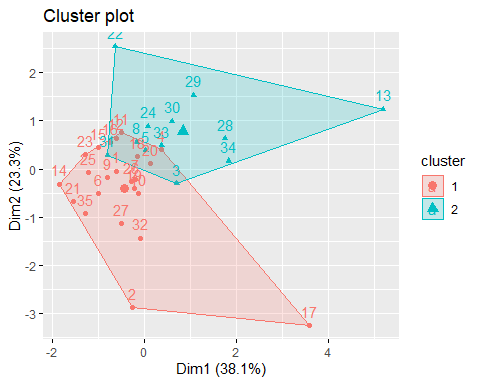
Looks like the optimal number is two. Ideally these will seperate strongly into groups that represent either island side or some combination of two of our site types.

Next for fun we will visualize our data as a scatter cluster plot.

add\_cluster\_groups <- cluster\_ready %>% dplyr::mutate(cluster = sub\_group\_dendro\_1)  
  
cluster\_site <- agnes(add\_cluster\_groups, method = "average")

## Warning in data.matrix(x): NAs introduced by coercion

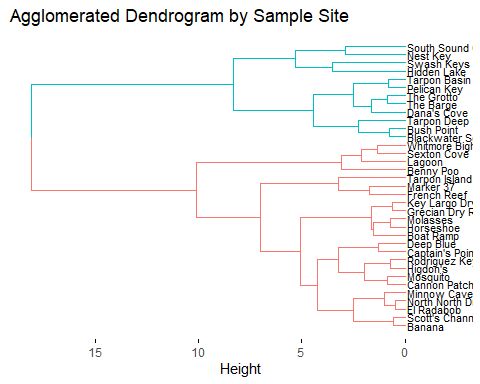
cluster\_data <- add\_cluster\_groups[, c(2:6)]  
  
#Uncomment to add names to plot (confusing in this situation)  
#rownames(cluster\_data) <- add\_cluster\_groups$location  
  
  
  
fviz\_cluster(list(data = cluster\_data, cluster = sub\_group\_dendro\_1))



Looks like most of our data is relatively close but two reasonalbly distinct clusters are present.

Next lets make a nice dendrogram.

library(factoextra)  
  
sample\_site\_dendro\_agnes <- fviz\_dend(cluster\_site\_agnes, k = 2, show\_labels = TRUE, cex = 0.6, rect = FALSE, color\_labels\_by\_k = FALSE, horiz = TRUE, rect\_lty = 5, rect\_border = 8, main = "Agglomerated Dendrogram by Sample Site")  
  
sample\_site\_dendro\_agnes



#Uncomment to preduce the full dendrogram for the hclust model.  
#sample\_site\_dendro\_hclust <- fviz\_dend(cluster\_site\_hclust, k = 2, show\_labels = TRUE, cex = 0.6, rect = FALSE, color\_labels\_by\_k = FALSE, horiz = TRUE, rect\_lty = 5, rect\_border = 8, main = "Cluster Dendrogram by Sample Site")  
  
#sample\_site\_dendro\_hclust

Now that we have the overall dendrogram for each site lets compare our three site types individuall and see how different each of them are from one another. For this analysis we will use average values for each parameter as we did above and compare clustering for Seagrass/Mangrove, Patch Reef/Hardbottom, and Coral Reef sites respectively.

First we need to subset data from the clean data, select our predictors, remove any NAs in the predictors, and finally aggregate the means of each site.

location\_average <- aggregate(. ~ location, cluster\_trim, mean)  
  
  
sg\_mg\_subset <- subset(WQ\_clean\_data, site\_type == "Seagrass/Mangrove")  
pr\_h\_subset <- subset(WQ\_clean\_data, site\_type == "Patch Reef/Hardbottom")  
cr\_subset <- subset(WQ\_clean\_data, site\_type == "Coral Reef")  
  
sg\_mg\_reduce <- sg\_mg\_subset %>% dplyr::select(location, water\_temp, salinity, dissolved\_oxygen, ammonia, ph)  
pr\_h\_reduce <- pr\_h\_subset %>% dplyr::select(location, water\_temp, salinity, dissolved\_oxygen, ammonia, ph)  
cr\_reduce <- cr\_subset %>% dplyr::select(location, water\_temp, salinity, dissolved\_oxygen, ammonia, ph)  
  
sg\_mg\_ready <- na.omit(sg\_mg\_reduce)  
pr\_h\_ready <- na.omit(pr\_h\_reduce)  
cr\_ready <- na.omit(cr\_reduce)  
  
sg\_mg\_ready <- aggregate(. ~ location, sg\_mg\_ready, mean)  
pr\_h\_ready <- aggregate(. ~ location, pr\_h\_ready, mean)  
cr\_ready <- aggregate(. ~ location, cr\_ready, mean)  
  
rownames(sg\_mg\_ready) <- sg\_mg\_ready$location  
rownames(pr\_h\_ready) <- pr\_h\_ready$location  
rownames(cr\_ready) <- cr\_ready$location

We will drop the Buttonwood observation again as the temperture value is very likely an error.

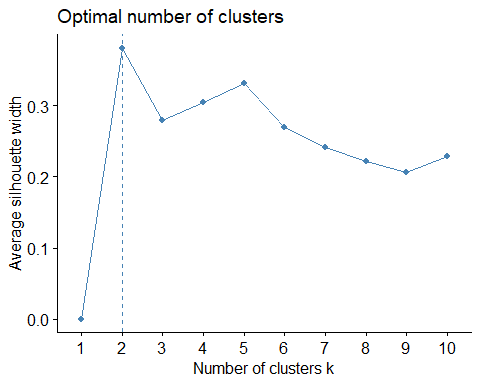
sg\_mg\_ready <- sg\_mg\_ready[!sg\_mg\_ready$location == "Buttonwood", ]

Now we can make our dendrograms, we will start with Seagrass/Mangrove.

fviz\_nbclust(sg\_mg\_ready, FUN = hcut, method = "silhouette")

## Warning in stats::dist(x): NAs introduced by coercion

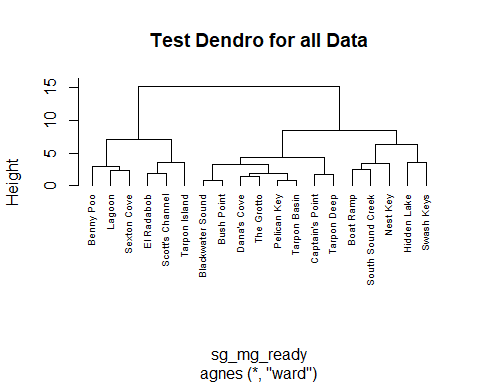
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

 Two clusters it is!

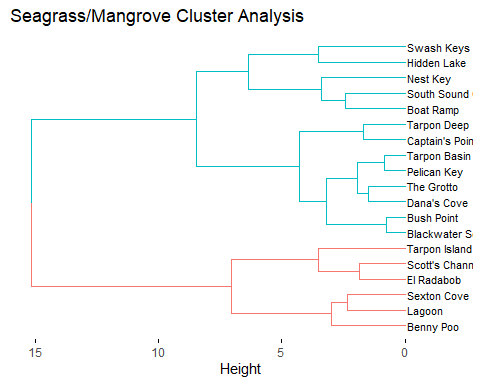
set.seed(1111)  
  
sg\_mg\_clust <- agnes(sg\_mg\_ready, method = "ward")

## Warning in data.matrix(x): NAs introduced by coercion

sg\_mg\_dend <- pltree(sg\_mg\_clust, cex = 0.6, hang = -1, main = "Test Dendro for all Data", labels = sg\_mg\_ready$location)



sg\_mg\_good\_dend <- fviz\_dend(sg\_mg\_clust, k = 2, show\_labels = TRUE, cex = 0.6, rect = FALSE, color\_labels\_by\_k = FALSE, horiz = TRUE, rect\_lty = 5, rect\_border = 8, main = "Seagrass/Mangrove Cluster Analysis")  
  
sg\_mg\_good\_dend

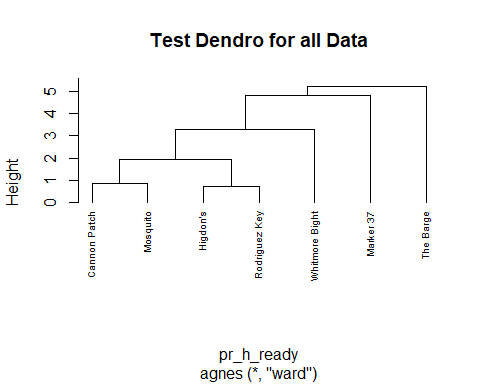
 Interesting! It appears that our clustering is reasonably consistent with geographic location. The red cluster successfully grouped all of the sites that are components of Largo Sound (an individual body of water). The blue cluster is a bit more dynamic but has the vast majority of the bayside sites (excluding Tarpon Island and Sexton Cove). A vast majority of the nearest neigbors in the dendrogram additionally are sites that are adjacent or very close to one another.

Lets take a look at patch reefs and hardbottom sites.

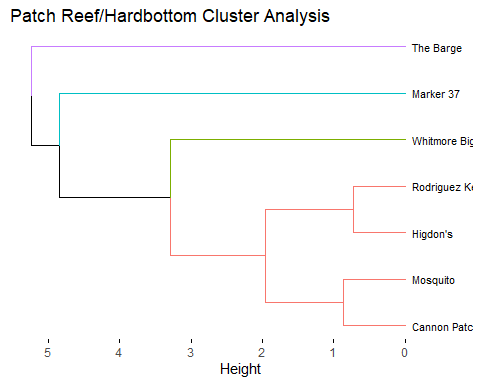
set.seed(1111)  
  
pr\_h\_clust <- agnes(pr\_h\_ready, method = "ward")

## Warning in data.matrix(x): NAs introduced by coercion

pr\_h\_dend <- pltree(pr\_h\_clust, cex = 0.6, hang = -1, main = "Test Dendro for all Data", labels = pr\_h\_ready$location)



pr\_h\_good\_dend <- fviz\_dend(pr\_h\_clust, k = 4, show\_labels = TRUE, cex = 0.6, rect = FALSE, color\_labels\_by\_k = FALSE, horiz = TRUE, rect\_lty = 5, rect\_border = 8, main = "Patch Reef/Hardbottom Cluster Analysis")  
  
pr\_h\_good\_dend



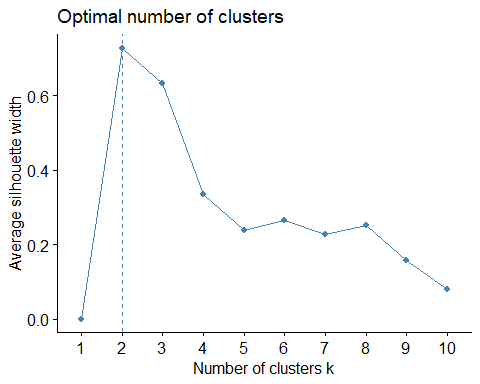
Looks like our Patch Reef/Hardbottom sites are quite different, this is not terribly suprising due to the fact these locations are quite variable depending on site and they are visited less frequently. Of note Cannon Patch and Mosquito are sister reefs (very similar structure/biodiversity and close in proximity) so it is interesting thst they are clustered closely. Whitemore Bight, Marker 37, and The Barge are very different sites from all other locations so it is unsuprising they are clustered independently.

Lastly lets investigate Coral Reefs.

fviz\_nbclust(cr\_ready, FUN = hcut, method = "silhouette")

## Warning in stats::dist(x): NAs introduced by coercion

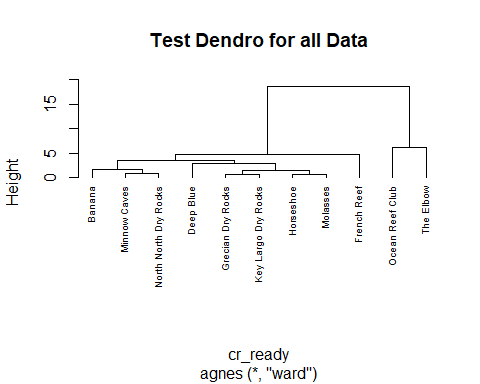
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

 Two clusters again.

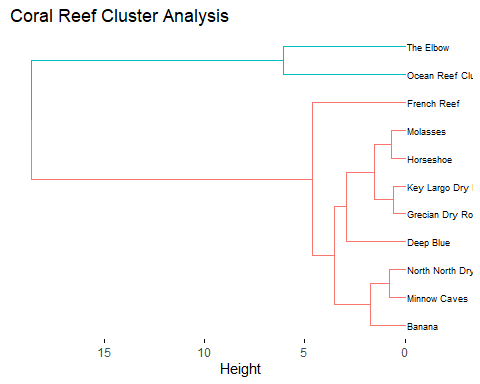
set.seed(1111)  
  
cr\_clust <- agnes(cr\_ready, method = "ward")

## Warning in data.matrix(x): NAs introduced by coercion

cr\_dend <- pltree(cr\_clust, cex = 0.6, hang = -1, main = "Test Dendro for all Data", labels = cr\_ready$location)



cr\_good\_dend <- fviz\_dend(cr\_clust, k = 2, show\_labels = TRUE, cex = 0.5, rect = FALSE, color\_labels\_by\_k = FALSE, horiz = TRUE, rect\_lty = 5, rect\_border = 8, main = "Coral Reef Cluster Analysis")  
  
cr\_good\_dend

 Coral reef clustering shows a very interesting pattern. The blue cluster contains the two northernmost reef sites visited by MarineLab. The red cluster contains the mid-southern reefs predominantly. While there are many breaks in the dendrogram, individual pairs are consistent geographically for North North Dry Rocks & Minow Caves, Key Largo Dry Rocks & Grecian Dry Rock. Both of these pairs represent adjacent reef systems.

Last lets make a comparison across years using a tanglegram to see how clustering of invividual site locations compares across the ~4 years of study.

First we must subset our data into years and normalize our locations so we only have repedative locations for each dendrogram.

year\_16 <- subset(WQ\_clean\_data, Year == "16")  
year\_17 <- subset(WQ\_clean\_data, Year == "17")  
year\_18 <- subset(WQ\_clean\_data, Year == "18")  
year\_19 <- subset(WQ\_clean\_data, Year == "19")  
  
year\_16 <- na.omit(year\_16)  
year\_17 <- na.omit(year\_17)  
year\_18 <- na.omit(year\_18)  
year\_19 <- na.omit(year\_19)  
  
year\_16 <- year\_16 %>% dplyr::select(location, water\_temp, salinity, dissolved\_oxygen, ammonia, ph)  
year\_17 <- year\_17 %>% dplyr::select(location, water\_temp, salinity, dissolved\_oxygen, ammonia, ph)  
year\_18 <- year\_18 %>% dplyr::select(location, water\_temp, salinity, dissolved\_oxygen, ammonia, ph)  
year\_19 <- year\_19 %>% dplyr::select(location, water\_temp, salinity, dissolved\_oxygen, ammonia, ph)  
  
year\_16 <- aggregate(. ~ location, year\_16, mean)  
year\_17 <- aggregate(. ~ location, year\_17, mean)  
year\_18 <- aggregate(. ~ location, year\_18, mean)  
year\_19 <- aggregate(. ~ location, year\_19, mean)  
  
  
rownames(year\_16) <- year\_16$location  
rownames(year\_17) <- year\_17$location  
rownames(year\_18) <- year\_18$location  
rownames(year\_19) <- year\_19$location

year\_16 <- year\_16[!year\_16$location == "Scott's Channel", ]  
year\_16 <- year\_16[!year\_16$location == "Sexton Cove", ]  
year\_16 <- year\_16[!year\_16$location == "Ocean Reef Club", ]  
year\_16 <- year\_16[!year\_16$location == "North North Dry Rocks", ]  
year\_16 <- year\_16[!year\_16$location == "NA", ]  
year\_16 <- year\_16[!year\_16$location == "Higdon's", ]  
year\_16 <- year\_16[!year\_16$location == "The Elbow", ]  
year\_16 <- year\_16[!year\_16$location == "The Barge", ]  
year\_16 <- year\_16[!year\_16$location == "Bayside", ]  
year\_16 <- year\_16[!year\_16$location == "Blackwater Sound", ]  
year\_16 <- year\_16[!year\_16$location == "Benny Poo", ]  
year\_16 <- year\_16[!year\_16$location == "French Reef", ]  
year\_16 <- year\_16[!year\_16$location == "Deep Blue", ]  
year\_16 <- year\_16[!year\_16$location == "Marker 37", ]  
year\_16 <- year\_16[!year\_16$location == "Lagoon", ]  
year\_16 <- year\_16[!year\_16$location == "Tarpon Deep", ]  
year\_16 <- year\_16[!year\_16$location == "Mosquito", ]  
  
unique(year\_16$location)

## [1] "Banana" "Boat Ramp" "Bush Point"   
## [4] "Cannon Patch" "Dana's Cove" "El Radabob"   
## [7] "Grecian Dry Rocks" "Horseshoe" "Key Largo Dry Rocks"  
## [10] "Minnow Caves" "Molasses" "Nest Key"   
## [13] "Pelican Key" "Rodriguez Key" "Tarpon Basin"   
## [16] "The Grotto" "Whitmore Bight"

year\_17 <- year\_17[!year\_17$location == "Scott's Channel", ]  
year\_17 <- year\_17[!year\_17$location == "Sexton Cove", ]  
year\_17 <- year\_17[!year\_17$location == "Ocean Reef Club", ]  
year\_17 <- year\_17[!year\_17$location == "North North Dry Rocks", ]  
year\_17 <- year\_17[!year\_17$location == "NA", ]  
year\_17 <- year\_17[!year\_17$location == "Higdon's", ]  
year\_17 <- year\_17[!year\_17$location == "The Elbow", ]  
year\_17 <- year\_17[!year\_17$location == "The Barge", ]  
year\_17 <- year\_17[!year\_17$location == "Bayside", ]  
year\_17 <- year\_17[!year\_17$location == "Blackwater Sound", ]  
year\_17 <- year\_17[!year\_17$location == "Benny Poo", ]  
year\_17 <- year\_17[!year\_17$location == "French Reef", ]  
year\_17 <- year\_17[!year\_17$location == "Deep Blue", ]  
year\_17 <- year\_17[!year\_17$location == "Marker 37", ]  
year\_17 <- year\_17[!year\_17$location == "Lagoon", ]  
year\_17 <- year\_17[!year\_17$location == "Tarpon Deep", ]  
year\_17 <- year\_17[!year\_17$location == "Mosquito", ]  
year\_17 <- year\_17[!year\_17$location == "Swash Keys", ]  
year\_17 <- year\_17[!year\_17$location == "Captain's Point", ]  
  
  
unique(year\_17$location)

## [1] "Banana" "Boat Ramp" "Bush Point"   
## [4] "Cannon Patch" "Dana's Cove" "El Radabob"   
## [7] "Grecian Dry Rocks" "Horseshoe" "Key Largo Dry Rocks"  
## [10] "Minnow Caves" "Molasses" "Nest Key"   
## [13] "Pelican Key" "Rodriguez Key" "Tarpon Basin"   
## [16] "The Grotto" "Whitmore Bight"

year\_18 <- year\_18[!year\_18$location == "Scott's Channel", ]  
year\_18 <- year\_18[!year\_18$location == "Sexton Cove", ]  
year\_18 <- year\_18[!year\_18$location == "Ocean Reef Club", ]  
year\_18 <- year\_18[!year\_18$location == "North North Dry Rocks", ]  
year\_18 <- year\_18[!year\_18$location == "NA", ]  
year\_18 <- year\_18[!year\_18$location == "Higdon's", ]  
year\_18 <- year\_18[!year\_18$location == "The Elbow", ]  
year\_18 <- year\_18[!year\_18$location == "The Barge", ]  
year\_18 <- year\_18[!year\_18$location == "Bayside", ]  
year\_18 <- year\_18[!year\_18$location == "Blackwater Sound", ]  
year\_18 <- year\_18[!year\_18$location == "Benny Poo", ]  
year\_18 <- year\_18[!year\_18$location == "French Reef", ]  
year\_18 <- year\_18[!year\_18$location == "Deep Blue", ]  
year\_18 <- year\_18[!year\_18$location == "Marker 37", ]  
year\_18 <- year\_18[!year\_18$location == "Lagoon", ]  
year\_18 <- year\_18[!year\_18$location == "Tarpon Deep", ]  
year\_18 <- year\_18[!year\_18$location == "Mosquito", ]  
year\_18 <- year\_18[!year\_18$location == "Swash Keys", ]  
year\_18 <- year\_18[!year\_18$location == "Buttonwood", ]  
year\_18 <- year\_18[!year\_18$location == "Captain's Point", ]  
  
  
unique(year\_18$location)

## [1] "Banana" "Boat Ramp" "Bush Point"   
## [4] "Cannon Patch" "Dana's Cove" "El Radabob"   
## [7] "Grecian Dry Rocks" "Horseshoe" "Key Largo Dry Rocks"  
## [10] "Minnow Caves" "Molasses" "Nest Key"   
## [13] "Pelican Key" "Rodriguez Key" "Tarpon Basin"   
## [16] "The Grotto" "Whitmore Bight"

year\_19 <- year\_19[!year\_19$location == "Scott's Channel", ]  
year\_19 <- year\_19[!year\_19$location == "Sexton Cove", ]  
year\_19 <- year\_19[!year\_19$location == "Ocean Reef Club", ]  
year\_19 <- year\_19[!year\_19$location == "North North Dry Rocks", ]  
year\_19 <- year\_19[!year\_19$location == "NA", ]  
year\_19 <- year\_19[!year\_19$location == "Higdon's", ]  
year\_19 <- year\_19[!year\_19$location == "The Elbow", ]  
year\_19 <- year\_19[!year\_19$location == "The Barge", ]  
year\_19 <- year\_19[!year\_19$location == "Bayside", ]  
year\_19 <- year\_19[!year\_19$location == "Blackwater Sound", ]  
year\_19 <- year\_19[!year\_19$location == "Benny Poo", ]  
year\_19 <- year\_19[!year\_19$location == "French Reef", ]  
year\_19 <- year\_19[!year\_19$location == "Deep Blue", ]  
year\_19 <- year\_19[!year\_19$location == "Marker 37", ]  
year\_19 <- year\_19[!year\_19$location == "Lagoon", ]  
year\_19 <- year\_19[!year\_19$location == "Tarpon Deep", ]  
year\_19 <- year\_19[!year\_19$location == "Mosquito", ]  
year\_19 <- year\_19[!year\_19$location == "Swash Keys", ]  
year\_19 <- year\_19[!year\_19$location == "Buttonwood", ]  
year\_19 <- year\_19[!year\_19$location == "Tarpon Island", ]  
year\_19 <- year\_19[!year\_19$location == "South Sound Creek", ]  
year\_19 <- year\_19[!year\_19$location == "Hidden Lake", ]  
year\_19 <- year\_19[!year\_19$location == "Captain's Point", ]  
  
unique(year\_19$location)

## [1] "Banana" "Boat Ramp" "Bush Point"   
## [4] "Cannon Patch" "Dana's Cove" "El Radabob"   
## [7] "Grecian Dry Rocks" "Horseshoe" "Key Largo Dry Rocks"  
## [10] "Minnow Caves" "Molasses" "Nest Key"   
## [13] "Pelican Key" "Rodriguez Key" "Tarpon Basin"   
## [16] "The Grotto" "Whitmore Bight"

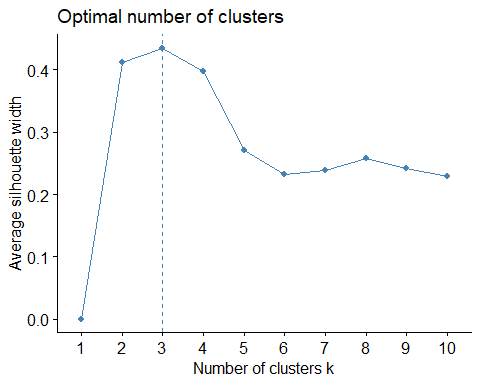
Phew lots of sorting, now we can start on our dendros.

We will make 2016 first.

fviz\_nbclust(year\_16, FUN = hcut, method = "silhouette")

## Warning in stats::dist(x): NAs introduced by coercion

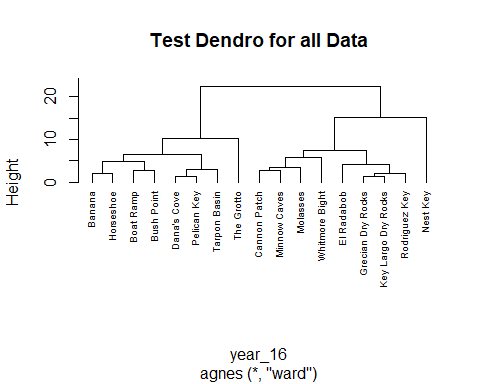
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion



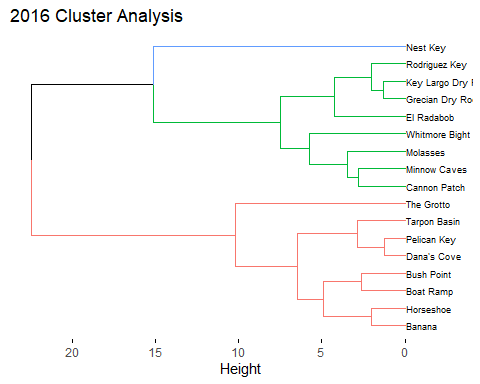
set.seed(1111)  
clust\_16 <- agnes(year\_16, method = "ward")

## Warning in data.matrix(x): NAs introduced by coercion

dend\_16 <- pltree(clust\_16, cex = 0.6, hang = -1, main = "Test Dendro for all Data", labels = year\_16$location)



good\_dend\_16 <- fviz\_dend(clust\_16, k = 3, show\_labels = TRUE, cex = 0.5, rect = FALSE, color\_labels\_by\_k = FALSE, horiz = TRUE, rect\_lty = 5, rect\_border = 8, main = "2016 Cluster Analysis")  
  
good\_dend\_16



dist\_16 <- dist(year\_16, method = "euclidean")

## Warning in dist(year\_16, method = "euclidean"): NAs introduced by coercion

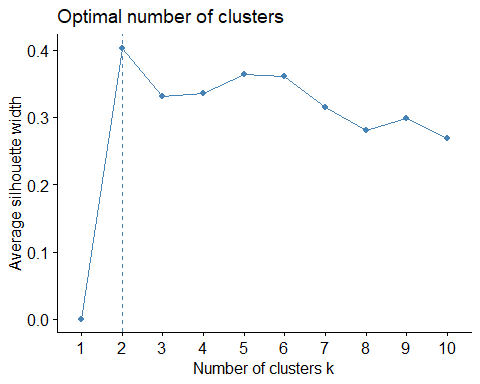
hclust\_16 <- hclust(dist\_16, method = "ward.D2")  
dend\_ready\_16 <- as.dendrogram(hclust\_16)

Now for 2017!

fviz\_nbclust(year\_17, FUN = hcut, method = "silhouette")

## Warning in stats::dist(x): NAs introduced by coercion

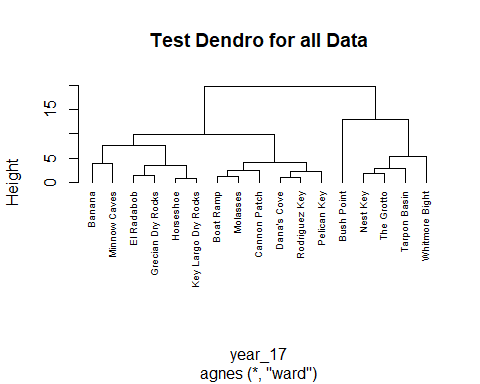
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion



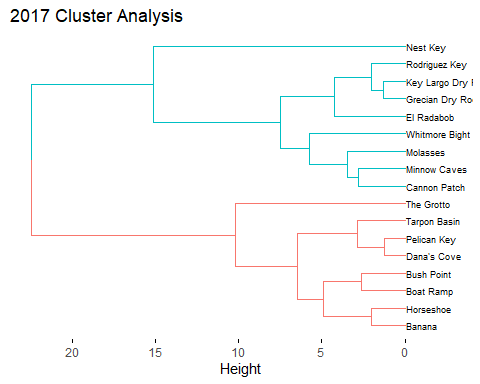
set.seed(1111)  
clust\_17 <- agnes(year\_17, method = "ward")

## Warning in data.matrix(x): NAs introduced by coercion

dend\_17 <- pltree(clust\_17, cex = 0.6, hang = -1, main = "Test Dendro for all Data", labels = year\_17$location)



good\_dend\_17 <- fviz\_dend(clust\_16, k = 2, show\_labels = TRUE, cex = 0.5, rect = FALSE, color\_labels\_by\_k = FALSE, horiz = TRUE, rect\_lty = 5, rect\_border = 8, main = "2017 Cluster Analysis")  
  
good\_dend\_17



dist\_17 <- dist(year\_17, method = "euclidean")

## Warning in dist(year\_17, method = "euclidean"): NAs introduced by coercion

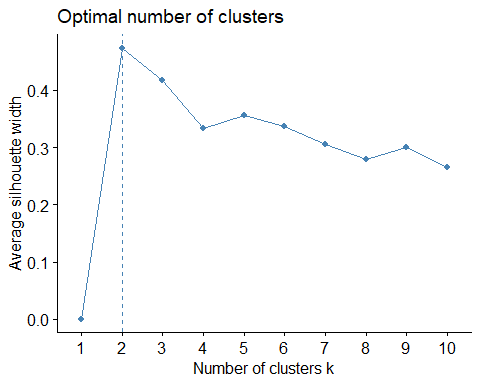
hclust\_17 <- hclust(dist\_17, method = "ward.D2")  
dend\_ready\_17 <- as.dendrogram(hclust\_17)

2018

fviz\_nbclust(year\_18, FUN = hcut, method = "silhouette")

## Warning in stats::dist(x): NAs introduced by coercion

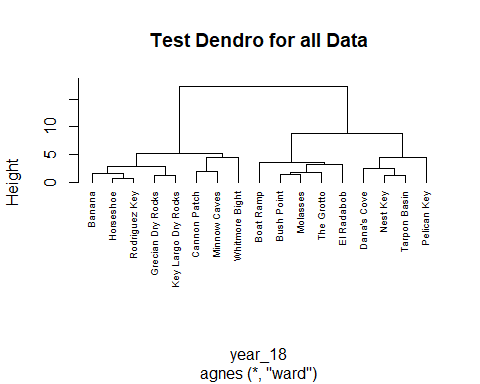
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion



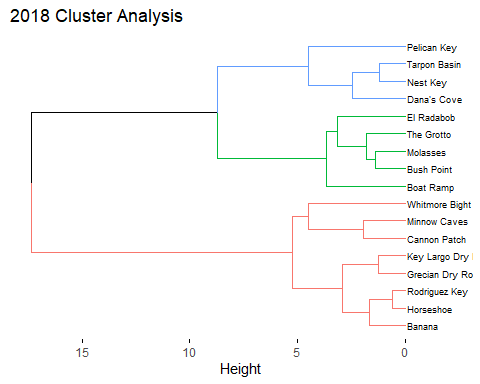
set.seed(1111)  
clust\_18 <- agnes(year\_18, method = "ward")

## Warning in data.matrix(x): NAs introduced by coercion

dend\_18 <- pltree(clust\_18, cex = 0.6, hang = -1, main = "Test Dendro for all Data", labels = year\_18$location)



good\_dend\_18 <- fviz\_dend(clust\_18, k = 3, show\_labels = TRUE, cex = 0.5, rect = FALSE, color\_labels\_by\_k = FALSE, horiz = TRUE, rect\_lty = 5, rect\_border = 8, main = "2018 Cluster Analysis")  
  
good\_dend\_18



dist\_18 <- dist(year\_18, method = "euclidean")

## Warning in dist(year\_18, method = "euclidean"): NAs introduced by coercion

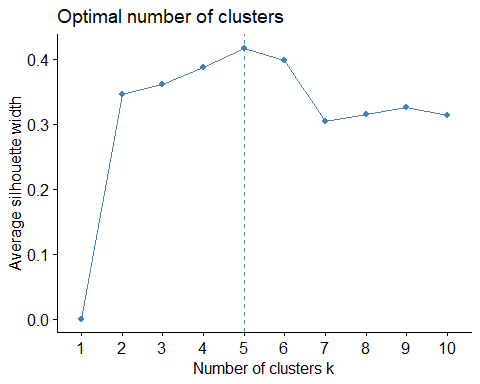
hclust\_18 <- hclust(dist\_18, method = "ward.D2")  
dend\_ready\_18 <- as.dendrogram(hclust\_18)

And 2019

fviz\_nbclust(year\_19, FUN = hcut, method = "silhouette")

## Warning in stats::dist(x): NAs introduced by coercion

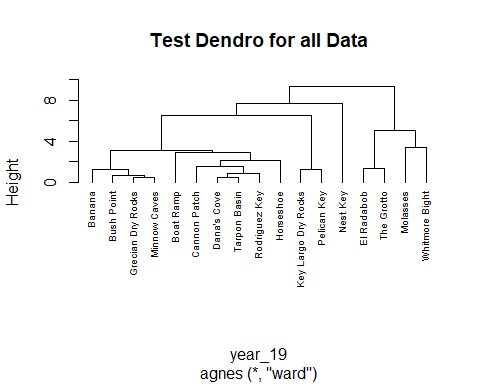
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion  
  
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion



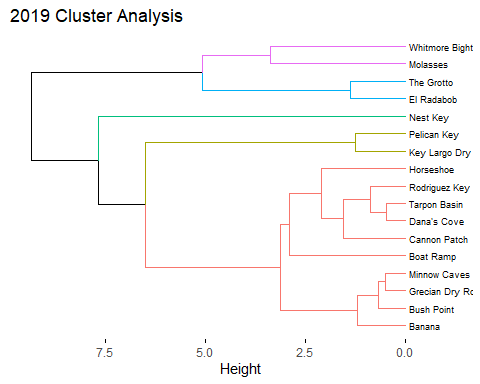
set.seed(1111)  
clust\_19 <- agnes(year\_19, method = "ward")

## Warning in data.matrix(x): NAs introduced by coercion

dend\_19 <- pltree(clust\_19, cex = 0.6, hang = -1, main = "Test Dendro for all Data", labels = year\_19$location)



good\_dend\_19 <- fviz\_dend(clust\_19, k = 5, show\_labels = TRUE, cex = 0.5, rect = FALSE, color\_labels\_by\_k = FALSE, horiz = TRUE, rect\_lty = 5, rect\_border = 8, main = "2019 Cluster Analysis")  
  
good\_dend\_19



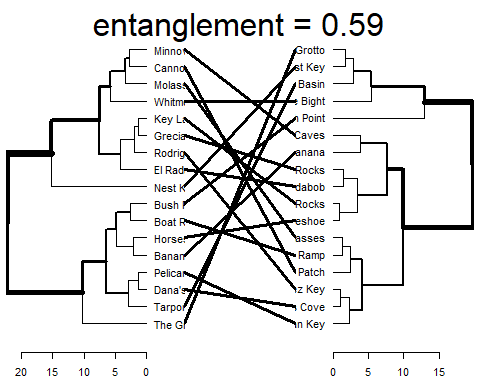
dist\_19 <- dist(year\_19, method = "euclidean")

## Warning in dist(year\_19, method = "euclidean"): NAs introduced by coercion

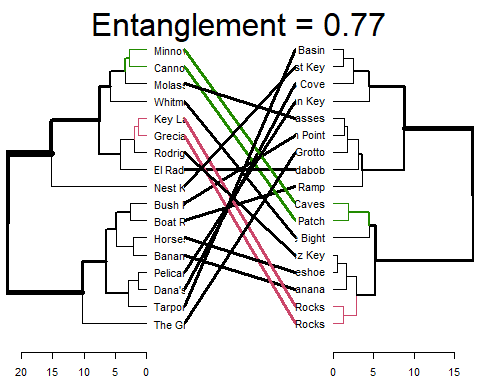
hclust\_19 <- hclust(dist\_19, method = "ward.D2")  
dend\_ready\_19 <- as.dendrogram(hclust\_19)

Now we can build our tanglegrams. Tanglegrams compare the alignment of two dendrograms. The entanglement coefficent represents the degree by which the two samples are entangled (or improperly aligned). A low entanglemnt shows a good alignment whereas a high entanglement shows disalignment.

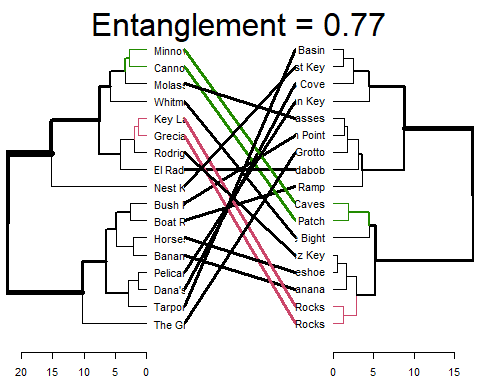
entang\_16\_17 <- dendlist(dend\_ready\_16, dend\_ready\_17)  
  
tangle\_16\_17 <- tanglegram(dend\_ready\_16, dend\_ready\_17, highlight\_distinct\_edges = FALSE, common\_subtrees\_color\_branches = TRUE, main = paste("entanglement =", round(entanglement(entang\_16\_17), 2)))



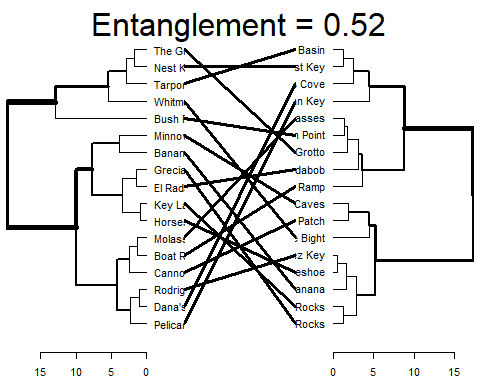
entang\_16\_18 <- dendlist(dend\_ready\_16, dend\_ready\_18)  
  
tangle\_16\_18 <- tanglegram(dend\_ready\_16, dend\_ready\_18, highlight\_distinct\_edges = FALSE, common\_subtrees\_color\_branches = TRUE, main = paste("Entanglement =", round(entanglement(entang\_16\_18), 2)))



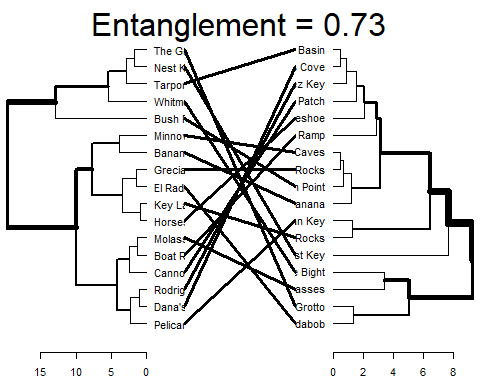
entang\_16\_18 <- dendlist(dend\_ready\_16, dend\_ready\_18)  
  
tangle\_16\_18 <- tanglegram(dend\_ready\_16, dend\_ready\_18, highlight\_distinct\_edges = FALSE, common\_subtrees\_color\_branches = TRUE, main = paste("Entanglement =", round(entanglement(entang\_16\_18), 2)))



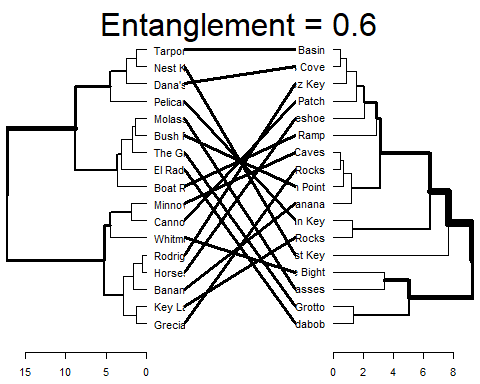
entang\_17\_18 <- dendlist(dend\_ready\_17, dend\_ready\_18)  
  
tangle\_17\_18 <- tanglegram(dend\_ready\_17, dend\_ready\_18, highlight\_distinct\_edges = FALSE, common\_subtrees\_color\_branches = TRUE, main = paste("Entanglement =", round(entanglement(entang\_17\_18), 2)))



entang\_17\_19 <- dendlist(dend\_ready\_17, dend\_ready\_19)  
  
tangle\_17\_19 <- tanglegram(dend\_ready\_17, dend\_ready\_19, highlight\_distinct\_edges = FALSE, common\_subtrees\_color\_branches = TRUE, main = paste("Entanglement =", round(entanglement(entang\_17\_19), 2)))



entang\_18\_19 <- dendlist(dend\_ready\_18, dend\_ready\_19)  
  
tangle\_18\_19 <- tanglegram(dend\_ready\_18, dend\_ready\_19, highlight\_distinct\_edges = FALSE, common\_subtrees\_color\_branches = TRUE, main = paste("Entanglement =", round(entanglement(entang\_18\_19), 2)))



So it looks like our closest aligning pairs are 2016 to 2017, 2017 to 2019 and 2018 to 2019. The closer alignment of consecutive years my reflect changes in overseeing staff over time or perhaps modification or sampling protocols. Regarding the best entanglement 2017 to 2018, the measure of 0.52 is not terribly robust. Therefore ther emay be some room for improvement to establish a more percise measuring protocol in future years. That being said citizen science data is typically highly variable thus a strong alignment was not expected, however it would be interesting to see if the staff could optimize methods from 2019 to 2020.