Statistical Methods for Ecology With The Hawaii Ocean Time-series (HOT)

Boopalakrishnan Arul

2024-01-28

Introduction

Since 1998, the University of Hawai'i at Manoa has maintained the Hawaii Ocean Time-series, a continuous record of ecological and physicochemical measurements taken at varying depths in a location north of Oahu. This notebook will look at a subset of that data: the counts per milliliter (# / mL) of four categories of microorganisms, as well as some relevant physical and chemical variables (depth, salinity). These cells were counted with flow cytometry.

The categories of microorganisms surveyed are: - Prochlorococcus, a genus of cyanobacteria. They are among the most abundant primary producers on earth. - Synechococcus, another genus of cyanobacteria. - Autotrophic eukaryotes, in particular pico-eukaryotes (3 micrometers or less in size). - Heterotrophic bacteria and archaea. Since these are not as dependent on light, they are found in abundance at depths (< 100 m) where numbers of the other three categories taper off rapidly.

More information at: https://hahana.soest.hawaii.edu/hot/methods/bact.html Describes basic details about the microorganism counts data. https://hahana.soest.hawaii.edu/hot/protocols/protocols.html?Chapter=15 For more in depth converage of the methods. https://hahana.soest.hawaii.edu/hot/methods/pprod.html Describes the primary productivity data which the bacterial counts data are included with. https://hahana.soest.hawaii.edu/FTP/hot/primary_production/Readme.pp Table specification of the dataset files.

Inspection of Data

```
data <- read.csv2("processed_data.csv")
data[1:5,]
```

```
##   CruiseID       Date Depth Salinity Prochlorococcus HeterotrophBacteria
## 1      319 2020-01-30     5  34.7482          211321               474969
## 2      319 2020-01-30    25  34.7705          211540               470114
## 3      319 2020-01-30    45  34.7832          226089               455934
## 4      319 2020-01-30    75  34.8938          138842               393944
## 5      319 2020-01-30   100  35.0325           78312               327226
##   Synechococcus PicoEukaryotes Year Month Day
## 1          2438           1462 2020     1  30
## 2          2593           1666 2020     1  30
## 3          3125           2306 2020     1  30
## 4           844           2439 2020     1  30
## 5            52           3418 2020     1  30
```

The data in this notebook is a subset of the total HOT bacterial counts series, comprising 160 measurements taken from January 2020 to September 2022, at depths from 5 to 175 meters.

```
unique(data$Date)
```

```
##  [1] "2020-01-30" "2020-07-14" "2020-08-07" "2020-09-02" "2020-09-26"
##  [6] "2020-11-18" "2020-12-18" "2021-01-12" "2021-02-16" "2021-03-23"
## [11] "2021-04-13" "2021-05-16" "2021-06-22" "2021-07-17" "2021-10-29"
## [16] "2022-03-27" "2022-05-26" "2022-07-09" "2022-07-30" "2022-09-01"
```

```
range(data$Depth)
```

```
## [1]   5 175
```

Before creating a distance matrix from these observations, we can first 1) drop empty rows from the dataset 2) visually inspect the dataset for any patterns of periodicity or correlation.

Empty cells in the dataset are marked with a "-9". Dropping any row containing these will allow us to retain 151 observations.

```
categories <- c("Prochlorococcus", "HeterotrophBacteria", "Synechococcus", "PicoEukaryotes")
counts.matrix <- data[,categories]
retain.rows <- apply(counts.matrix, MARGIN = 1,
                     FUN = {function(row) all(row != c(-9, -9, -9, -9)) })

#drop all rows with a -9, signifying data that wasn't recorded
retained.data <- data[retain.rows,]
nrow(retained.data)
```

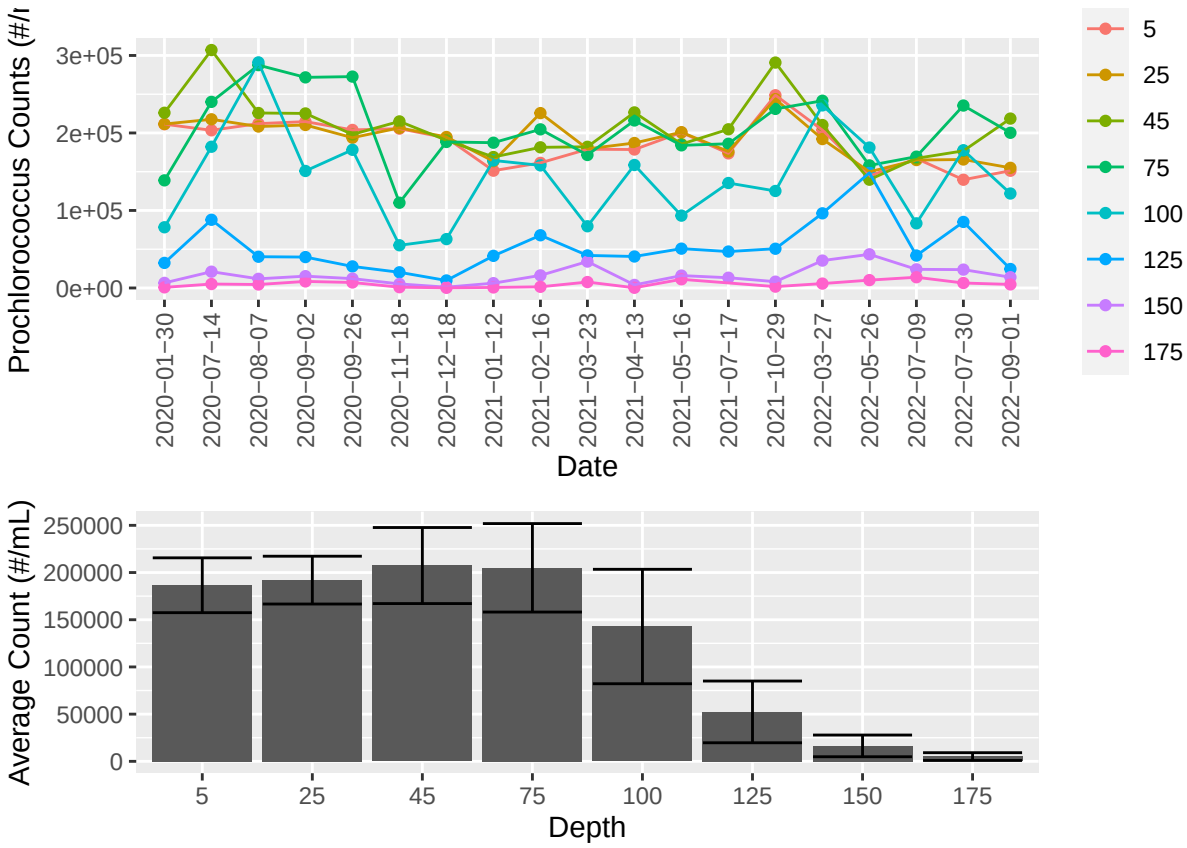
```
## [1] 151
```

This first series of plots will show the change in counts in each category over time.

```
library(ggplot2)
library(patchwork)

summary.plots <- function (category) {
  line.plot <- ggplot(retained.data, aes(x = Date, y = retained.data[,category],
                                         group = Depth,
                                         col=as.factor(Depth))) +
    geom_point() +
    geom_line() +
    scale_y_continuous(name = paste(category, "Counts (#/mL)")) +
    scale_color_discrete(name = "Depth") +
    scale_x_discrete(guide = guide_axis(angle = 90))

  #bar plot that shows mean count with 1 SD error bars for each depth subset
  #of each datum
  depths <- unique(retained.data$Depth)
  depths.summary <- data.frame(
    Depth = depths,
    Means = sapply(depths,
```

```
                    {function(depth)
                      mean(retained.data[retained.data$Depth == depth, category])}),
    Stdevs = sapply(depths,
                    {function(depth)
                      sd(retained.data[retained.data$Depth == depth, category])})
  )

  bar.plot <- ggplot(depths.summary, aes(x = as.factor(Depth), y = Means,
                                         ymin = Means - Stdevs,
                                         ymax = Means + Stdevs)) +
    geom_bar(stat="identity") +
    geom_errorbar() +
    scale_x_discrete(name = "Depth") +
    scale_y_continuous(name = "Average Count (#/mL)")
  print(line.plot + bar.plot + plot_layout(ncol = 1, heights = c(1,1)))
  depths.summary$CoefVariation <- depths.summary$Stdevs / depths.summary$Means
  print(depths.summary)
}
```

```
summary.plots("Prochlorococcus")
```

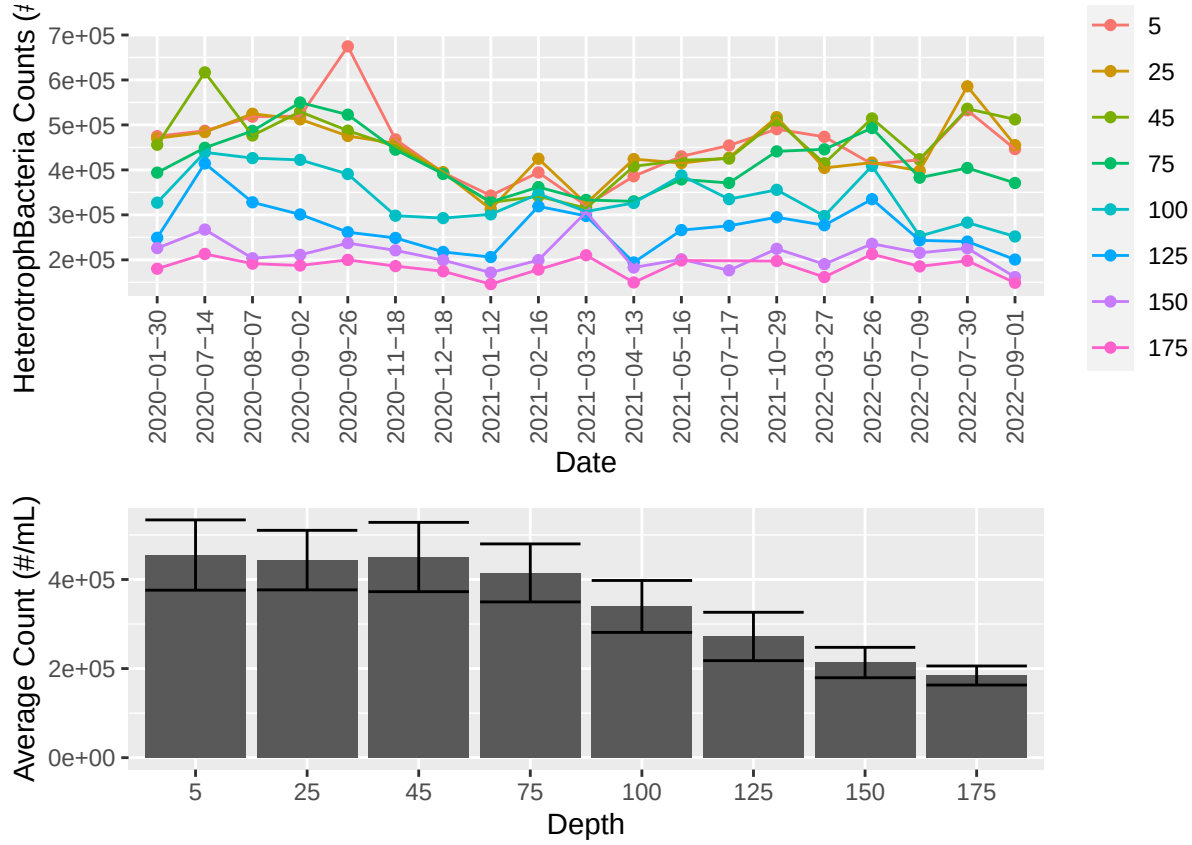



```
##   Depth      Means   Stdevs CoefVariation
## 1     5 186530.842 29027.94     0.1556201
## 2    25 191970.421 25321.48     0.1319030
## 3    45 207406.526 40277.83     0.1941975
```

```
## 4    75 205007.737 46866.80      0.2286099
## 5   100 142822.842 60606.08      0.4243445
## 6   125  52367.579 32690.42      0.6242492
## 7   150  16364.105 11493.03      0.7023316
## 8   175   5027.778  4124.55      0.8203525
```

As expected of a photoautotroph, Prochlorococcus is found in higher numbers at shallow depths. For the depths in which Prochlorococcus is abundant, the coefficient of variation is 0.25 or less. Starting at 100 meters, the counts are generally lower but subject to proportionally greater swings.

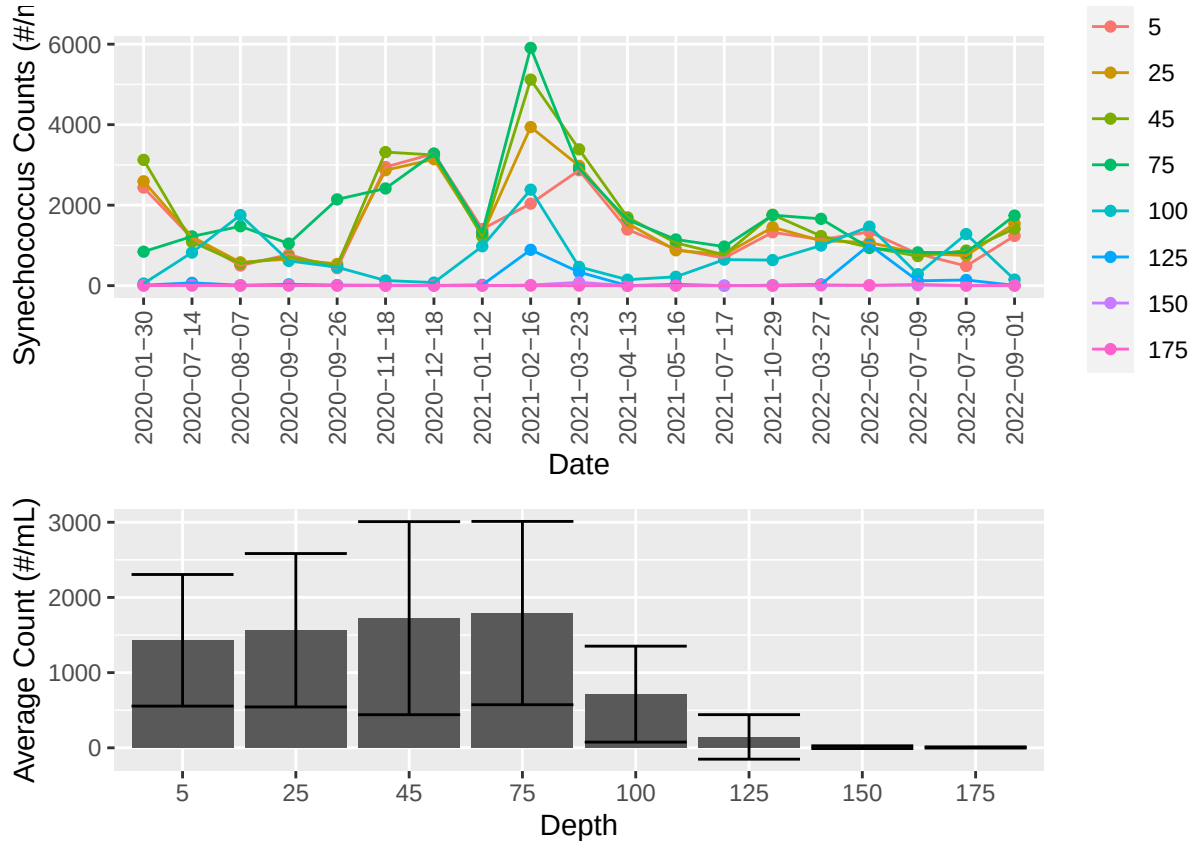
```
summary.plots("HeterotrophBacteria")
```




```
##   Depth    Means   Stdevs CoefVariation
## 1     5 454889.8 78859.33     0.1733592
## 2    25 443511.8 66837.68     0.1507010
## 3    45 450442.3 77777.82     0.1726699
## 4    75 414739.6 65099.76     0.1569654
## 5   100 339477.2 58308.04     0.1717584
## 6   125 272098.2 54302.65     0.1995701
## 7   150 213537.8 34132.74     0.1598440
## 8   175 184430.9 21356.65     0.1157976
```

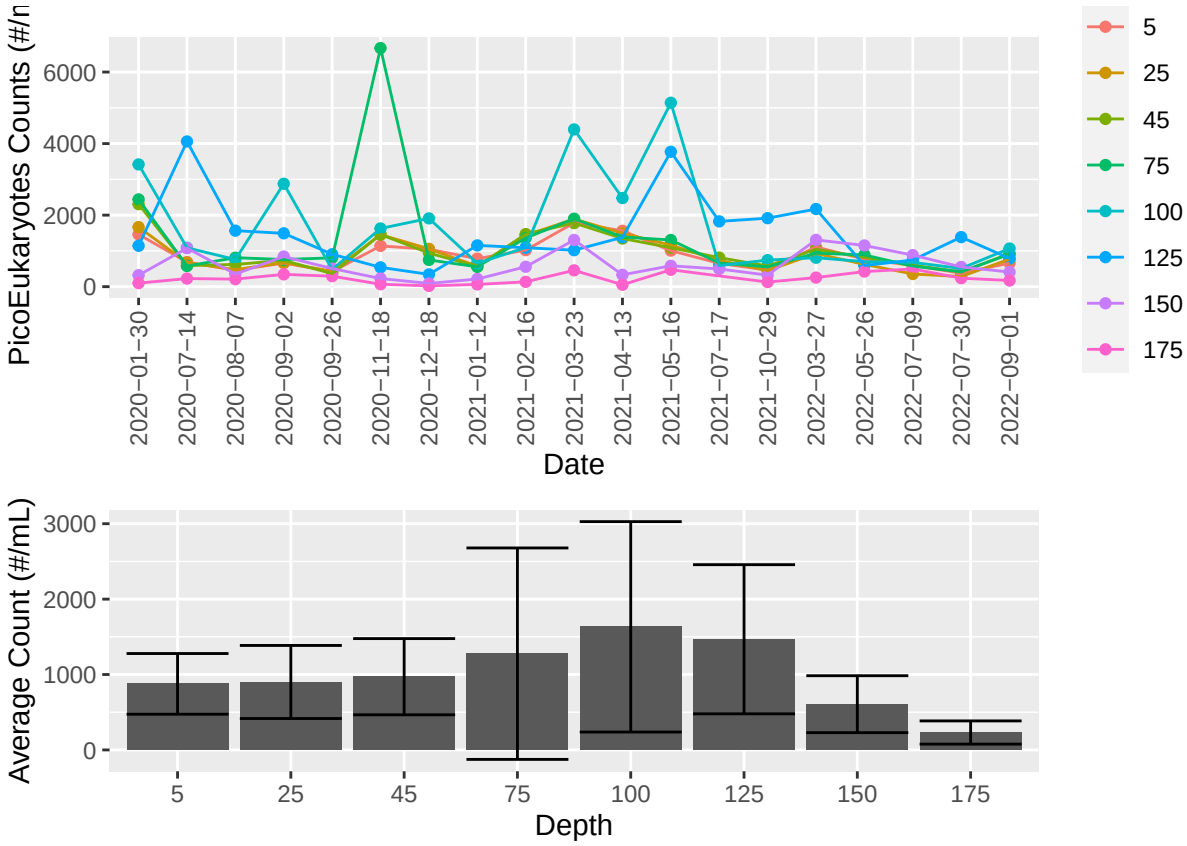
Heterotrophic microorganisms are the largest of the four categories, usually making up the majority of each sample in the timeseries. Standard deviations that would be considered large in the other three categories are proportionally no greater than 20% of the respective mean.

```
summary.plots("Synechococcus")
```




```
##   Depth       Means      Stdevs CoefVariation
## 1     5 1429.894737  875.382830     0.6122009
## 2    25 1563.947368 1020.203600     0.6523260
## 3    45 1724.368421 1284.037435     0.7446422
## 4    75 1792.368421 1219.162154     0.6801962
## 5   100  713.894737  638.064512     0.8937795
## 6   125  144.842105  296.107763     2.0443487
## 7   150    8.263158   18.217336     2.2046458
## 8   175    4.277778    8.027933     1.8766597
```

```
summary.plots("PicoEukaryotes")
```




```
##   Depth     Means    Stdevs CoefVariation
## 1     5  875.9474  402.4862     0.4594868
## 2    25  901.0000  484.6252     0.5378748
## 3    45  971.0526  505.3520     0.5204167
## 4    75 1276.4211 1402.1602     1.0985092
## 5   100 1632.4211 1395.2479     0.8547108
## 6   125 1467.6316  989.1382     0.6739690
## 7   150  606.6316  377.5881     0.6224340
## 8   175  230.7222  153.8065     0.6666307
```

Synechococcus and autotrophic eukaryotes are much less common than the other two categories of cells, and have the greatest coefficients of variation. Of note here are relative spikes in eukaryote counts in late 2020 and throughout 2021, in the middle depth range of 75 to 125 meters. These spikes, along with a spike in Synechococcus throughout the water column in early 2021, coincide with a lack of similar peaks in Prochlorococcus and heterotrophs.

Autotrophs not only compete with each other for nutrients, but also for light. As the cell count in the water grows, the turbidity or opacity of the water column will increase, and less light will penetrate through the water. It's possible that reduced competition from the rival Prochlorococcus allowed Synechococcus and small eukaryotes to grow in numbers.

Distance Matrices and Ordination

We could express the earlier observations about the autotrophic counts as a question: whether the samples taken in 2021, as a result of their distribution of the four categories, are really different from the samples

taken in the preceding and succeeding year. We can cluster samples by their similarity or dissimilarity to each other by making a distance matrix, and applying an ordination technique like nonmetric multidimensional scaling (nMDS).

```
library(vegan)
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.6-4
```

```
retained.counts <- retained.data[,categories]
relative.abundances <- decostand(retained.counts, method="total", MARGIN = 1)
distance.matrix <- as.matrix( vegdist(relative.abundances, method="chisq"), labels = T)
distance.matrix[1:6,1:6]
```

```
##           1         2         3        4         5         6
## 1 0.00000000 0.01066193 0.06666671 0.1336277 0.3157917 0.4722281
## 2 0.01066193 0.00000000 0.05716418 0.1386231 0.3199035 0.4784735
## 3 0.06666671 0.05716418 0.00000000 0.1867414 0.3640328 0.5305844
## 4 0.13362775 0.13862309 0.18674140 0.0000000 0.1834479 0.3511714
## 5 0.31579167 0.31990349 0.36403283 0.1834479 0.0000000 0.2080845
## 6 0.47222808 0.47847354 0.53058442 0.3511714 0.2080845 0.0000000
```

We will use Chi-squared distance to create the distance matrix, after first scaling all the counts so that they represent the relative abundance of each category. We use this distance instead of the more popular Bray-Curtis distance because we want differences in the shape of the distribution, such as an "unusual" relative rise in a previously rare category, to have a greater impact on the "distance" between two points. Bray-Curtis, a more common dissimilarity metric, is less sensitive to this.

```
nonmetric.scaling <- metaMDS(distance.matrix, distance = "chisq", k = 2)
```

```
## Run 0 stress 0.01420269
## Run 1 stress 0.02237012
## Run 2 stress 0.031205
## Run 3 stress 0.02782928
## Run 4 stress 0.02556189
## Run 5 stress 0.02047327
## Run 6 stress 0.02846487
## Run 7 stress 0.02193435
## Run 8 stress 0.03134474
## Run 9 stress 0.0213519
## Run 10 stress 0.02921695
## Run 11 stress 0.02136751
## Run 12 stress 0.03134773
## Run 13 stress 0.02182806
## Run 14 stress 0.02989383
## Run 15 stress 0.02950785
## Run 16 stress 0.02975623
## Run 17 stress 0.0241266
```

```
## Run 18 stress 0.01448157
## ... Procrustes: rmse 0.01729909  max resid 0.09367314
## Run 19 stress 0.03181775
## Run 20 stress 0.02980247
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      2: no. of iterations >= maxit
##      4: stress ratio > sratmax
##     14: scale factor of the gradient < sfgrmin
```

Nonmetric multidimensional scaling is an iterative algorithm that maximizes the correlation between two rank-ordered distance matrices. The first of these matrices is the "real" distance matrix: the distances between our actual samples, each of which is represented as a point in a 4-dimensional space. The second of these matrices is the ordination distance matrix, which will be populated with corresponding points in only 2 dimensions.

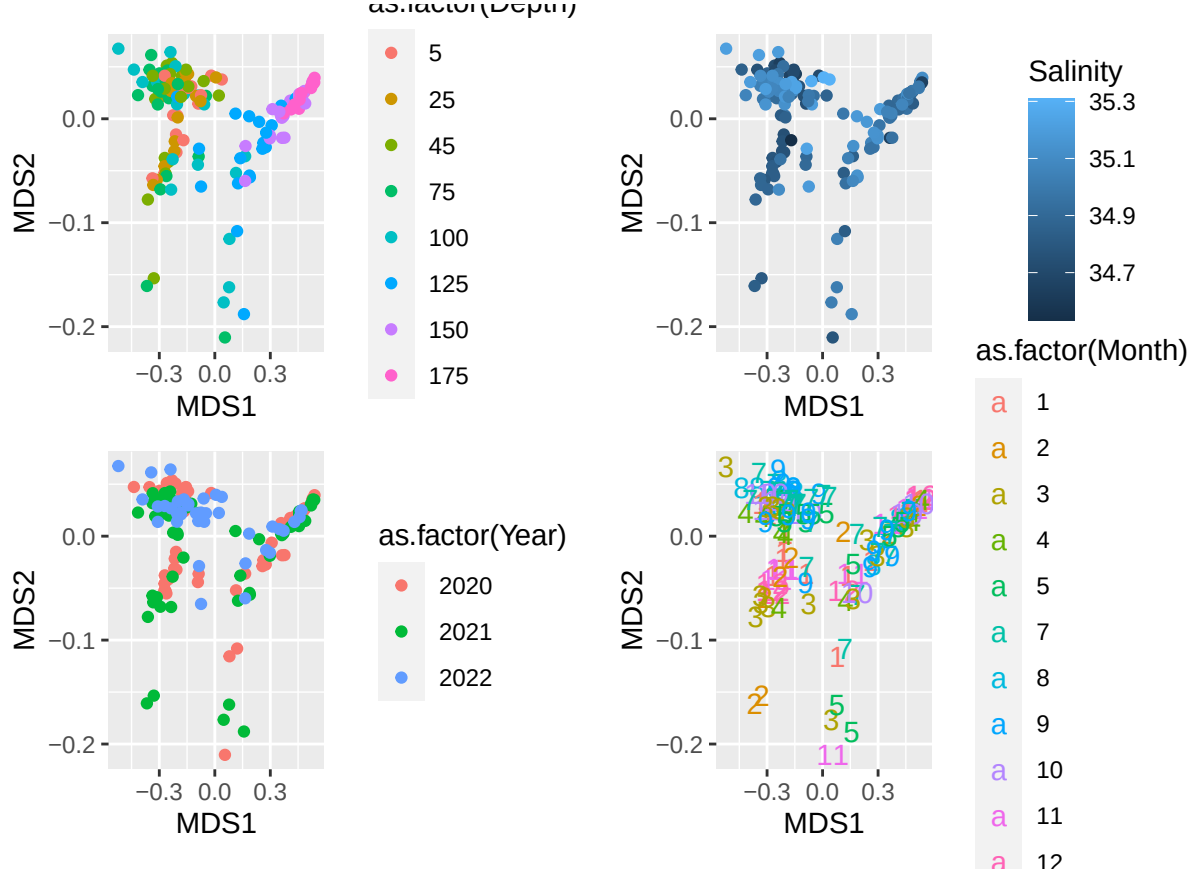
```
nonmetric.scaling$stress
```

```
## [1] 0.01420269
```

```
retained.data$MDS1 <- nonmetric.scaling$points[,"MDS1"]
retained.data$MDS2 <- nonmetric.scaling$points[,"MDS2"]
#stressplot(nonmetric.scaling)
```

The objective is for the rank ordered ordination distance matrix to correlate with the rank ordered real distance matrix. When this is achieved, the "stress" value will be low, ideally below 0.05. Since the stress here is around 0.02 (the algorithm is iterative, and may reach a slightly different solution each time), we will choose to keep the results of this ordination. We will add the new coordinate values, representing the position of each sample within the ordination space, to the data frame containing other useful values: depth, salinity, and date of sampling.

```
p1 <- ggplot(data = retained.data, aes(x = MDS1, y = MDS2, col = as.factor(Depth))) +
  geom_point()
p2 <- ggplot(data = retained.data, aes(x = MDS1, y = MDS2, col = Salinity)) +
  geom_point()
p3 <- ggplot(data = retained.data, aes(x = MDS1, y = MDS2, col = as.factor(Year))) +
  geom_point()
p4 <- ggplot(data = retained.data, aes(x = MDS1, y = MDS2, col = as.factor(Month))) +
  geom_text(data = retained.data, aes(label = Month))
p1 + p2 + p3 + p4 + plot_layout(ncol = 2)
```



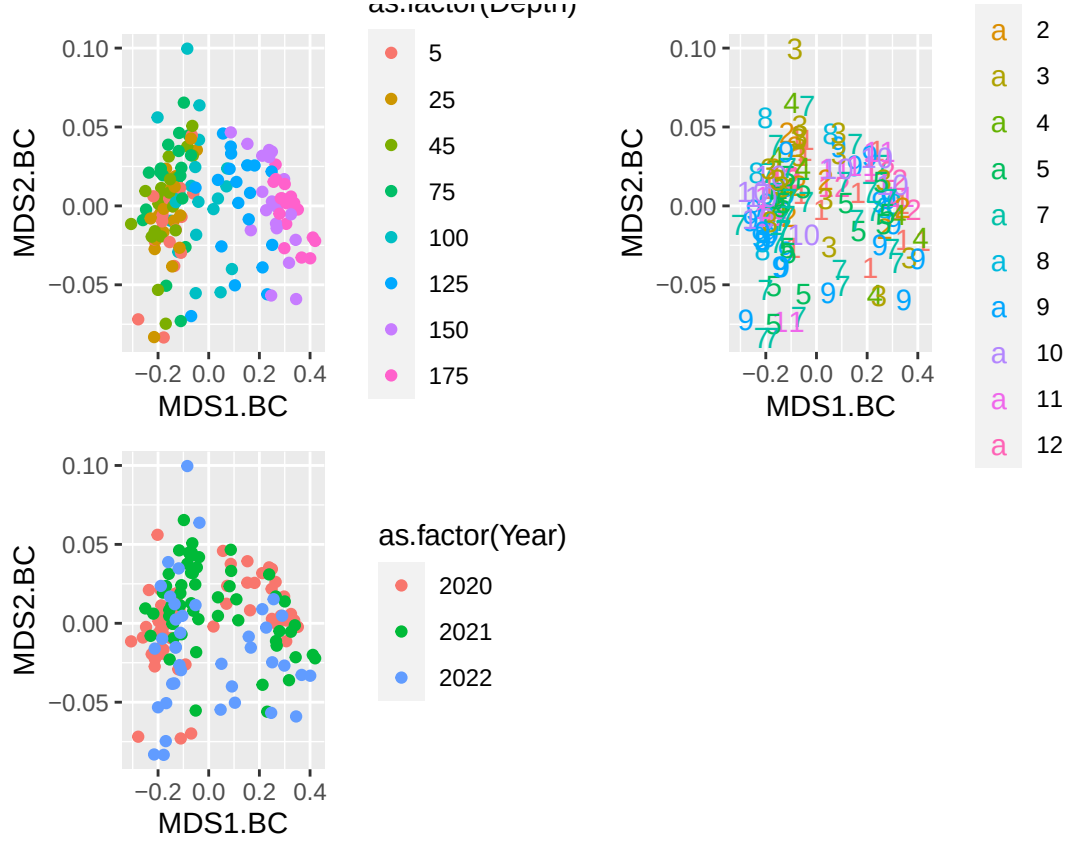
The above four plots all represent the same ordination space– the points are colored differently based on other properties associated with each sample. As can be seen here, the samples don't cluster particularly well by year, month, or salinity. They cluster best by depth, with samples of depth 75 and above on the left side of the graph and samples of depth 100 and below on the right side of the graph. Also of note is that samples from 2022 are, regardless of depth, associated with higher values along MDS2. It's possible that the year of sampling explains some variance.

```
distance.matrix.BC <- as.matrix( vegdist(retained.counts, method="bray"), labels = T)
nMDS.BC <- metaMDS(distance.matrix.BC, distance = "bray", k = 2, trace = 0)
nMDS.BC$stress
```

```
## [1] 0.02400911
```

```
retained.data$MDS1.BC <- nMDS.BC$points[,"MDS1"]
retained.data$MDS2.BC <- nMDS.BC$points[,"MDS2"]
```

```
p1 <- ggplot(data = retained.data, aes(x = MDS1.BC, y = MDS2.BC, col = as.factor(Depth))) +
  geom_point()
p2 <- ggplot(data = retained.data, aes(x = MDS1.BC, y = MDS2.BC, col = as.factor(Month))) +
  geom_text(label = retained.data$Month)
p3 <- ggplot(data = retained.data, aes(x = MDS1.BC, y = MDS2.BC, col = as.factor(Year))) +
  geom_point()
p1 + p2 + p3 + plot_layout(nrow = 2)
```



The creation of a new ordination, based on Bray-Curtis dissimilarity instead of chi-squared distance, supports the importance of depth for the distance between samples: again, samples of differing depths are clustered in distinct regions along MDS1. The observation of samples from 2022 clustering along higher values of MDS2, however, does not hold here.

As for which distance measure is "better", I would still go with the chi-squared for the reasons discussed above– however, the comparison does highlight that even if 2021's samples had a slightly different distribution of the two rarer categories, this change may not have been enough for these samples to be meaningfully distant from the other samples. In other words, the shape of a sample's distribution is largely determined by the most common categories within it, which are clearly the Prochlorococcus and the heterotrophic prokaryotes. As seen below, on average these two categories make up 99.6% of each sample.

```
mean(relative.abundances$Prochlorococcus + relative.abundances$HeterotrophBacteria)
```

```
## [1] 0.996137
```

There are two other methods we can use to better understand how samples at different depths differ from each other: the distance-based redundancy analysis (dbRDA) and the analysis of similarities (ANOSIM).

dbRDA may look similar to nMDS, since it also creates a graph meant to be interpreted by visual inspection for clusters or gradients. However, nMDS is an "unconstrained" ordination technique in which the only input for the procedure is the distance matrix created from the (relative, absolute) species abundances, and the output is meant only to reflect the input as closely as possible. It is only by coloring or otherwise marking these ordination-space points that we can try and guess at which variables (depth, salinity, etc) are most associated with particular clusters or gradients of points.

dbRDA is a "constrained" ordination method, with an additional input and objective. By treating the species-abundance distance matrix as a "response", dbRDA tries to relate it one or several explanatory variables.

dbRDA is a generalized version of redundancy analysis (RDA), and has a similar relationship to it as the lesser known "principal coordinates analysis" (PCoA, also known as metric or classical multidimensional scaling or MDS) has with the more famous principal components analysis (PCA). RDA and PCA also try to create lower-dimensional representations of a high-dimensional set of points, and both do so by preserving the relationship between high-dimensional distance and lower-dimension ordination distance. In the case of these methods, their definition of distance is only "Euclidean distance", the shortest path between two points in an n-dimensional space.

PCoA does for PCA what dbRDA does for RDA: allow the same method, but with a different definition of distance. This lets us use our chi-squared and Bray-Curtis distance matrices from earlier. In fact, a dbRDA is done by first applying a PCoA to an input distance matrix, and then applying an RDA to the results of that.

```
#left hand side of formula should be a distance/dissimilarity matrix created by vegdist
#data will contain the variables on right side of formula
chisq.dbRDA <- dbrda(formula = distance.matrix ~ Depth + Salinity + Year + Month,
                     data = retained.data)
chisq.dbRDA
```

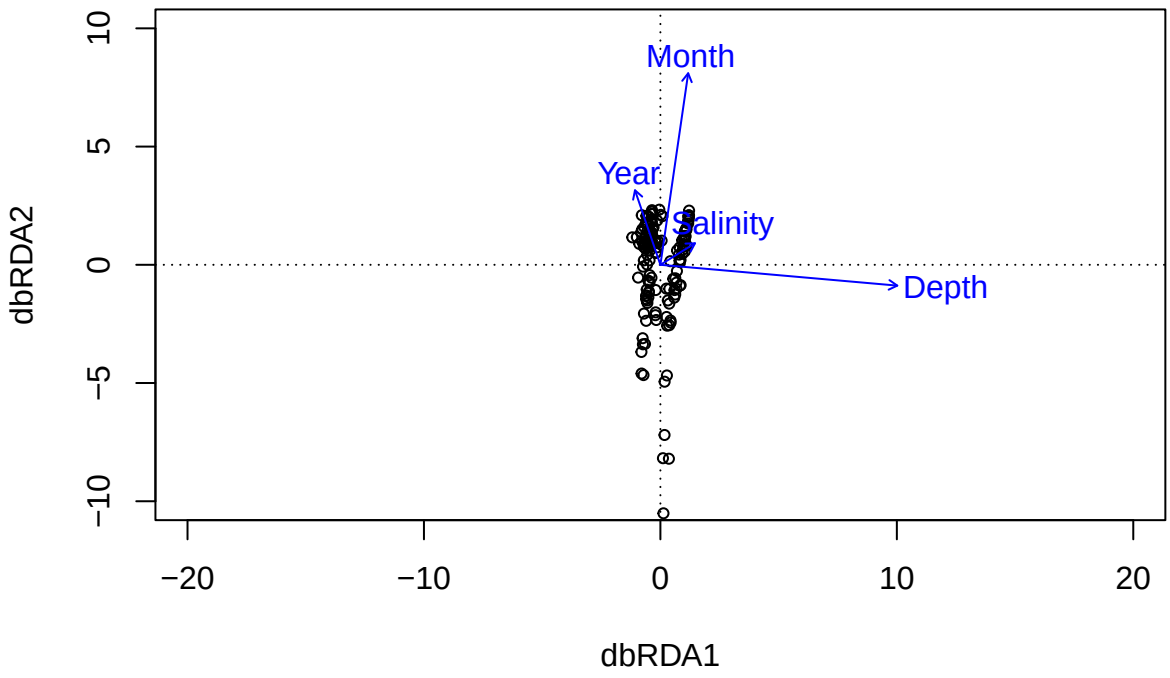
```
## Call: dbrda(formula = distance.matrix ~ Depth + Salinity + Year +
## Month, data = retained.data)
##
##               Inertia Proportion Rank RealDims
## Total         13.6915     1.0000
## Constrained    8.7924     0.6422    3        3
## Unconstrained  4.8991     0.3578    3        3
## Inertia is squared Unknown distance
##
## Eigenvalues for constrained axes:
## dbRDA1 dbRDA2 dbRDA3
##  8.764  0.018  0.010
##
## Eigenvalues for unconstrained axes:
##  MDS1  MDS2  MDS3
## 4.443 0.271 0.185
```

```
summary(chisq.dbRDA)$concont
```

```
## $importance
## Importance of components:
##                         dbRDA1   dbRDA2   dbRDA3
## Eigenvalue              8.7643 0.018134 0.009981
## Proportion Explained    0.9968 0.002062 0.001135
## Cumulative Proportion 0.9968 0.998865 1.000000
```

The most important values in this summary are the constrained and unconstrained proportions of the variance (second column). 64% of the variance is accounted for by these four explanatory variables. Next are the importance of each of the three components or axes produced by the dbRDA. dbRDA1 alone explains almost all of the variance.

```
plot(chisq.dbRDA, xlim = c(-10, 10), ylim = c(-10, 10))
```



dbRDA1 also turns out to be most parallel with the vector representing Depth's association with the dbRDA axes (a vector perpendicular to an axis could explain none of the variance along that axis; the opposite is true of a parallel vector). As can be seen below, even having Depth alone as the one explanatory variable would still explain 61% of the variance.

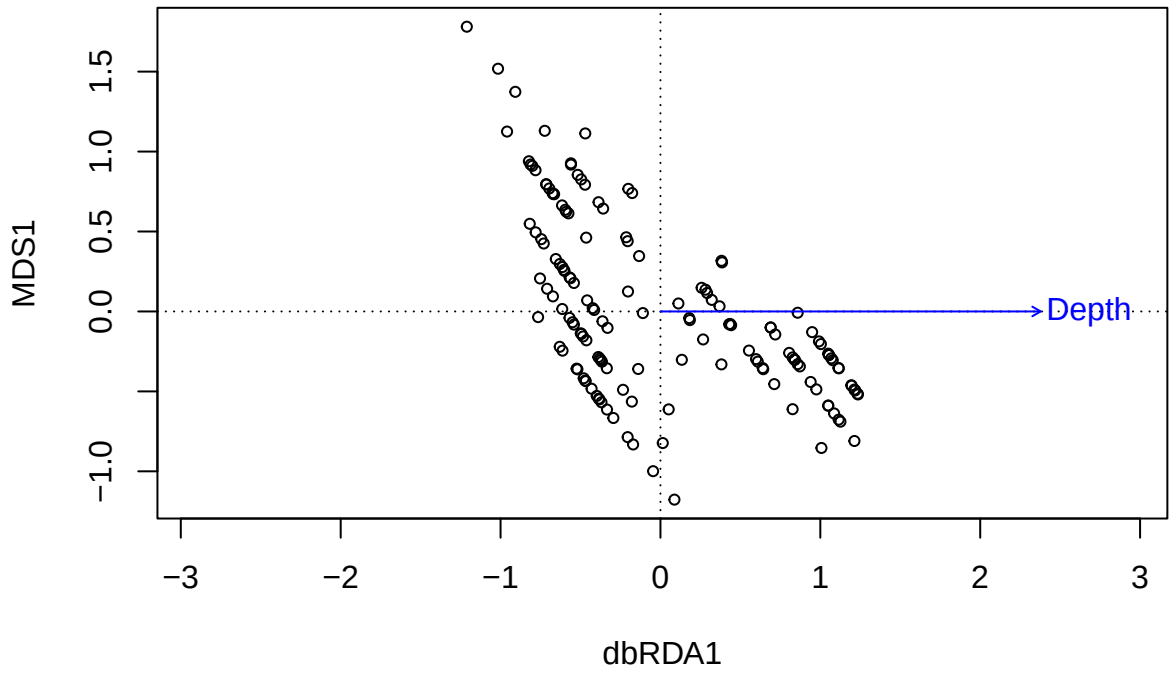
```
chisq.dbRDA.onevar <- dbrda(formula = distance.matrix ~ Depth,
                     data = retained.data)
chisq.dbRDA.onevar
```

```
## Call: dbrda(formula = distance.matrix ~ Depth, data = retained.data)
##
##               Inertia Proportion Rank
## Total         13.6915     1.0000
## Constrained    8.4463     0.6169    1
## Unconstrained  5.2452     0.3831    3
## Inertia is squared Unknown distance
##
## Eigenvalues for constrained axes:
## dbRDA1
##  8.446
##
## Eigenvalues for unconstrained axes:
##  MDS1  MDS2  MDS3
## 4.765 0.285 0.195
```

```r
summary(chisq.dbRDA.onevar)$concont
```

```
## $importance
## Importance of components:
##                         dbRDA1
## Eigenvalue              8.446
## Proportion Explained    1.000
## Cumulative Proportion   1.000
```

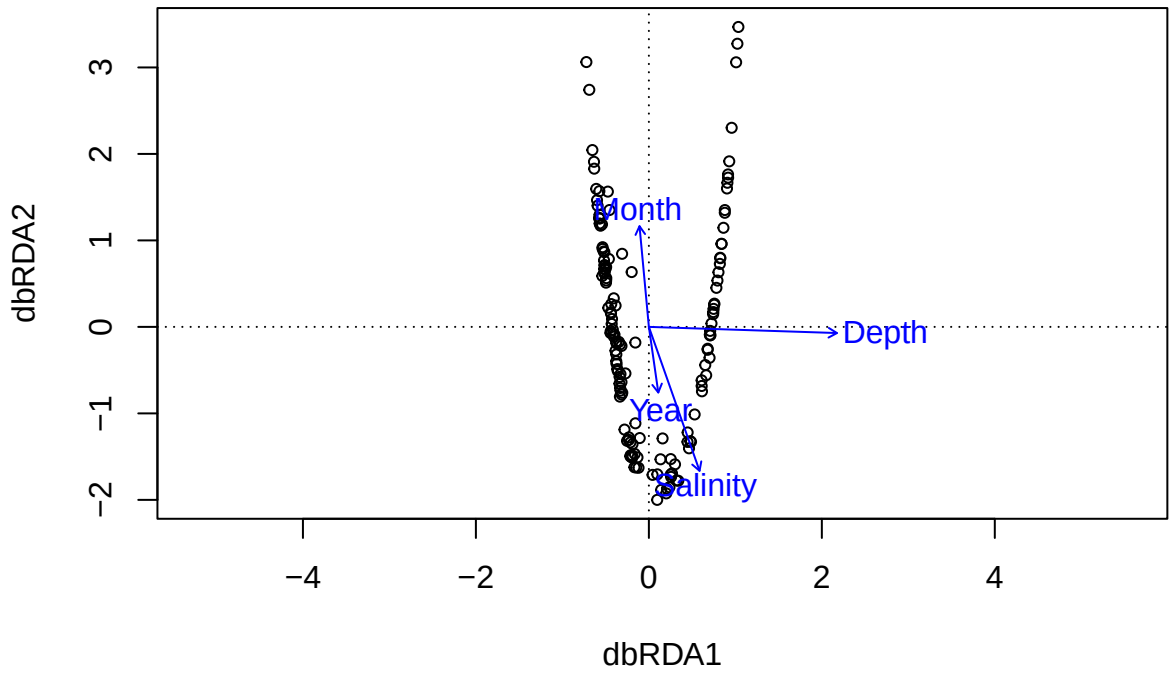
```r
plot(chisq.dbRDA.onevar)
```




```r
BC.dbRDA = dbrda(distance.matrix.BC ~ Depth + Salinity + Month + Year,
                 data = retained.data,
                 distance = "bray")
summary(BC.dbRDA)$concont
```

```
## $importance
## Importance of components:
##                         dbRDA1  dbRDA2   dbRDA3    dbRDA4
## Eigenvalue              5.7947 0.06199 0.008412 0.0025561
## Proportion Explained    0.9876 0.01057 0.001434 0.0004356
## Cumulative Proportion 0.9876 0.99813 0.999564 1.0000000
```

```
plot(BC.dbRDA)
```



The near-parabolic shape of the dbRDA based on Bray-Curtis distances is a surprise. The geometry of that looks interesting but I'm not sure how to explain that yet.

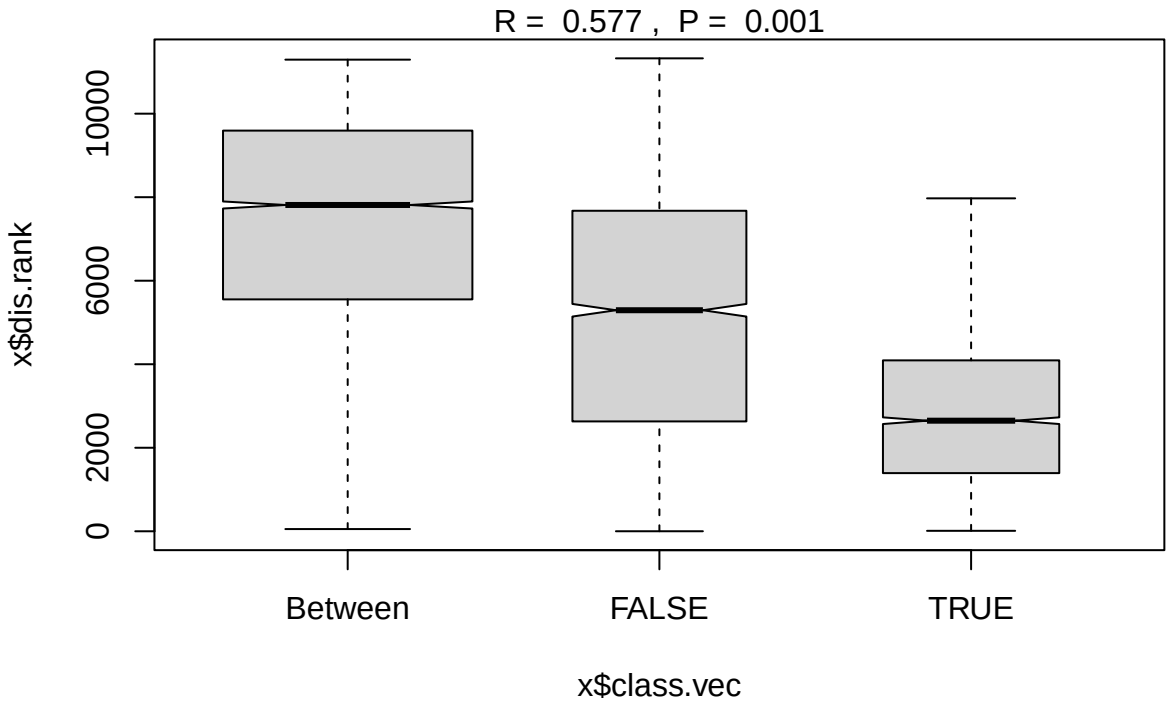
Distance Matrices and Hypothesis Testing

Analysis of Similarity (ANOSIM) is not an ordination/visualization technique, but a hypothesis test for statistically significant differences between groups. Imagine a line that runs between the points, separating them into two categories: for example, "shallow" (depth <= 75) and "deep" (depth >= 100).

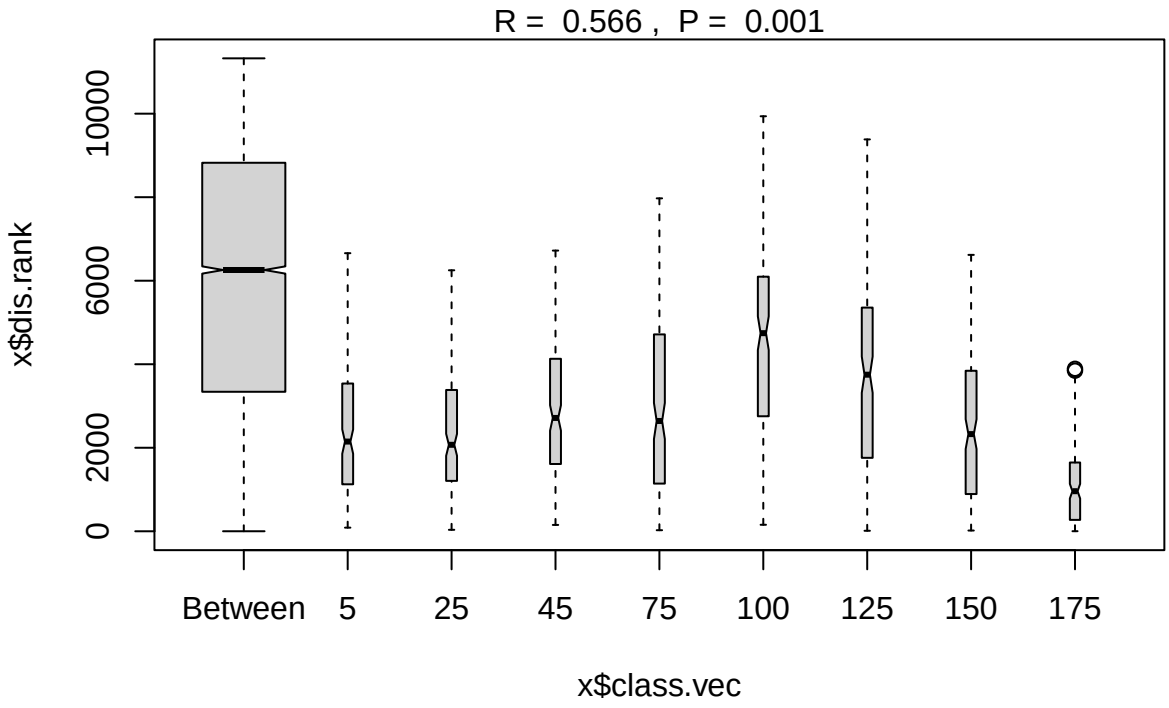
```
retained.data$DepthBinary <- retained.data$Depth <= 75
```

The ANOSIM will take the distance matrix and separate the distances into two categories: those that cross this imaginary line (distances "between" groups) and those that don't (distances "within" groups). All the distances are replaced by their "ranks" (the smallest is rank 1, and counting up from there). If the average rank of the within-group distances is smaller than the average rank of the between-group distances, and if this difference is "extreme" within the distribution of such size differences (a distribution obtained by running the algorithm hundreds of times with different permutations of the distances), then there is a statistically significant difference between the two groups of points.

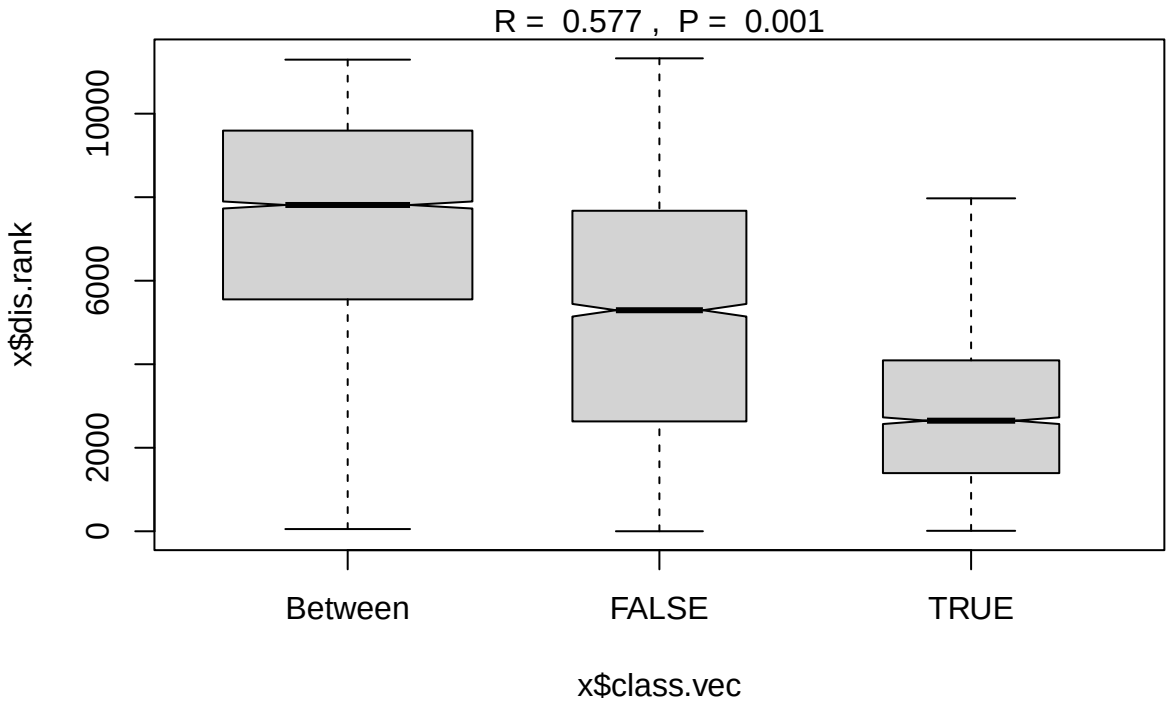
```
anosim.chisq.depthbinary <- anosim(distance.matrix, grouping = retained.data$DepthBinary, permutations =
plot(anosim.chisq.depthbinary)
```



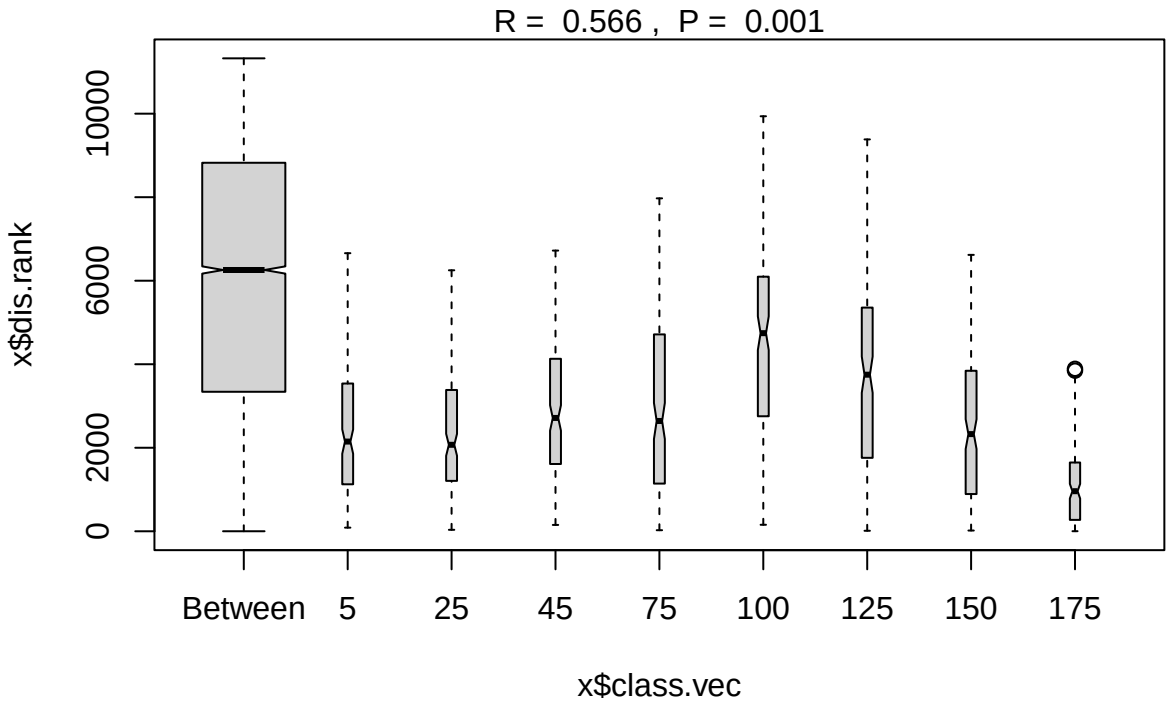

```
anosim.chisq.depthfactor <- anosim(distance.matrix, grouping = retained.data$Depth, permutations = 999)
plot(anosim.chisq.depthfactor)
```




```
anosim.BC.depthbinary <- anosim(distance.matrix, grouping = retained.data$DepthBinary, permutations = 99
plot(anosim.BC.depthbinary)
```

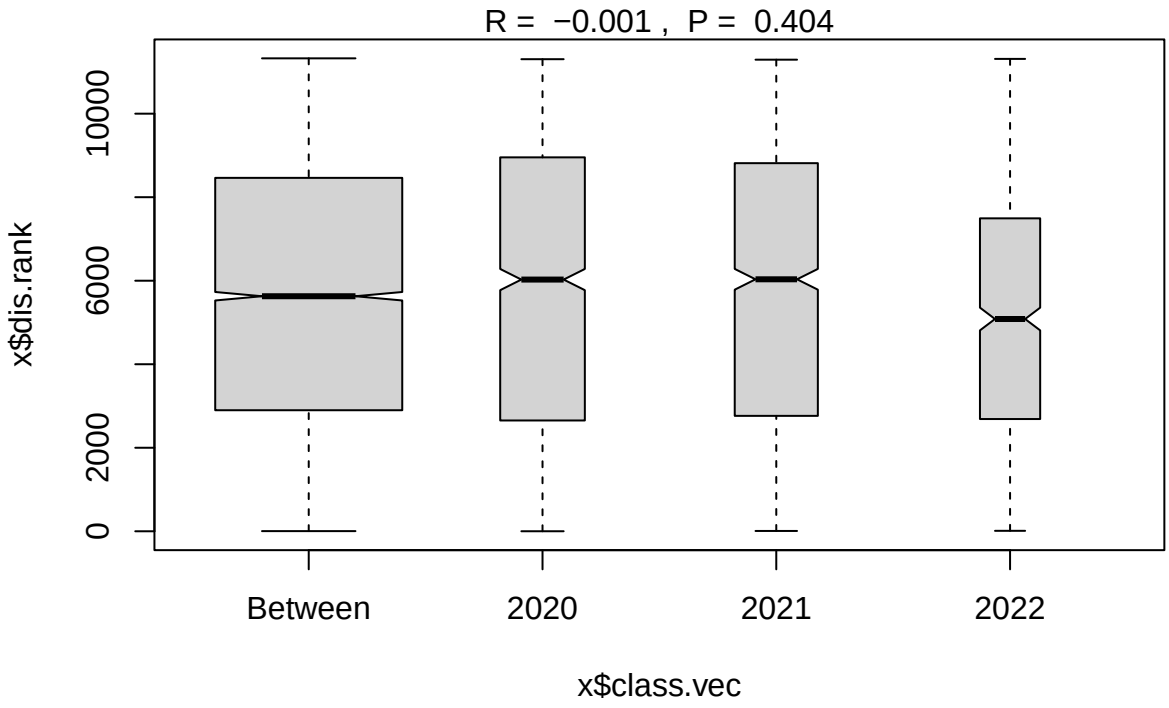



```
anosim.BC.depthfactor <- anosim(distance.matrix, grouping = retained.data$Depth, permutations = 999)
plot(anosim.BC.depthfactor)
```

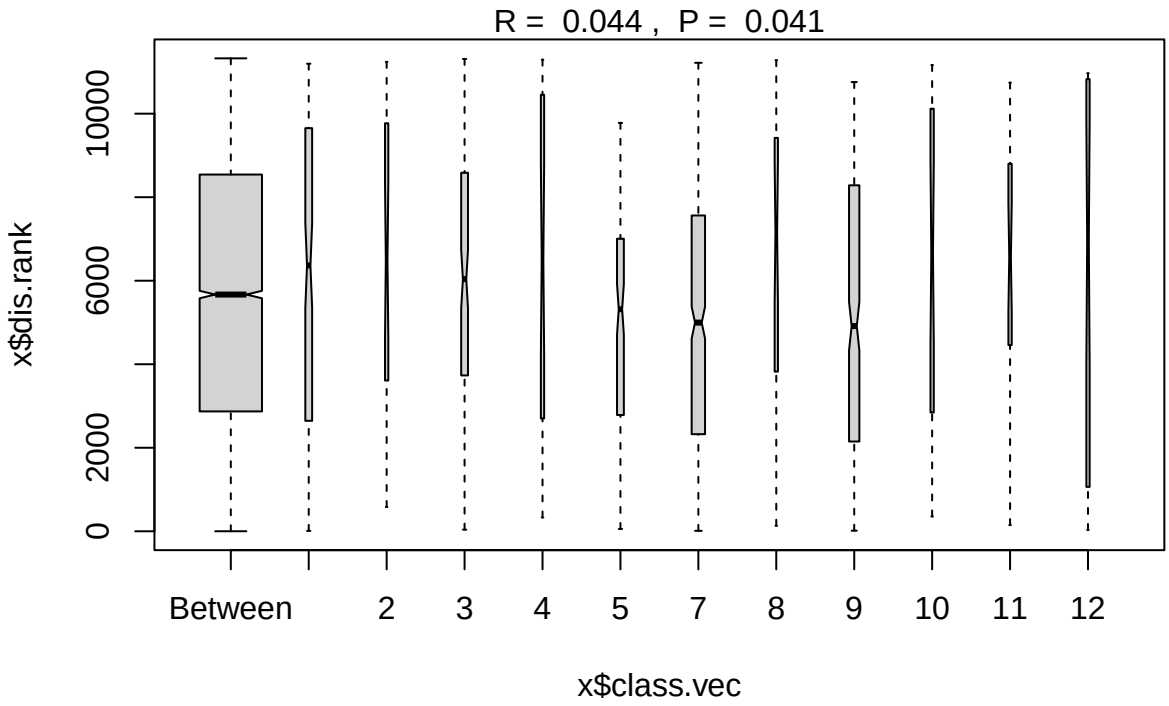


The test statistic "R" returned by ANOSIM is -1 when all the lowest-rank differences are between groups, 0 when there is a mix of high- and low- rank differences within (and between) groups, and 1 when the lowest-rank / smallest differences are all within groups. Whether depth is a binary factor or represented by all eight levels in the original dataset, and whether Bray Curtis or chi-squared distances are used, the ANOSIM calculates R to be above 0.5, a result with a p-value of 0.001. We can therefore conclude that the differences between samples taken at different depths are statistically significant.

```
anosim.chisq.yearfactor <- anosim(distance.matrix, grouping = retained.data$Year,
                                  permutations = 999)
plot(anosim.chisq.yearfactor)
```




```
anosim.chisq.monthfactor <- anosim(distance.matrix, grouping = retained.data$Month,
                                   permutations = 999)
plot(anosim.chisq.monthfactor)
```



Grouping by the year of sampling does not produce groups of samples with statistically significant differences. The month of sampling has a more impressive result, but this may be because there are too many groups for ANOSIM to handle effectively and too few samples within each group (there are only 151 samples retained by this experiment).

Conclusion

The Hawaiian Ocean Time-series turned out to be an interesting preview of challenges in this field of study. It may not be uncommon to find other environments where the distribution of a sample is largely dominated by very few categories of organisms, in turn making the explanatory variables that most affect those organisms' distributions into the variables that best explain the sample overall. The nMDS and dbRDA illustrated what the ANOSIM confirmed: that the top layer of the ocean, where so much of the world's primary production takes place, is itself a collection of diverse environments defined by availability of light, nutrients, and other rudiments of life.

References and Helpful Resources

https://chrischizinski.github.io/rstats/vegan-ggplot2/

https://uw.pressbooks.pub/appliedmultivariatestatistics/chapter/rda-and-dbrda/

https://uw.pressbooks.pub/appliedmultivariatestatistics/chapter/anosim/

https://rdrr.io/cran/vegan/man/capscale.html

Footnote

```
shallow.data <- retained.data[retained.data$Depth <= 75,]
dim(shallow.data)
```

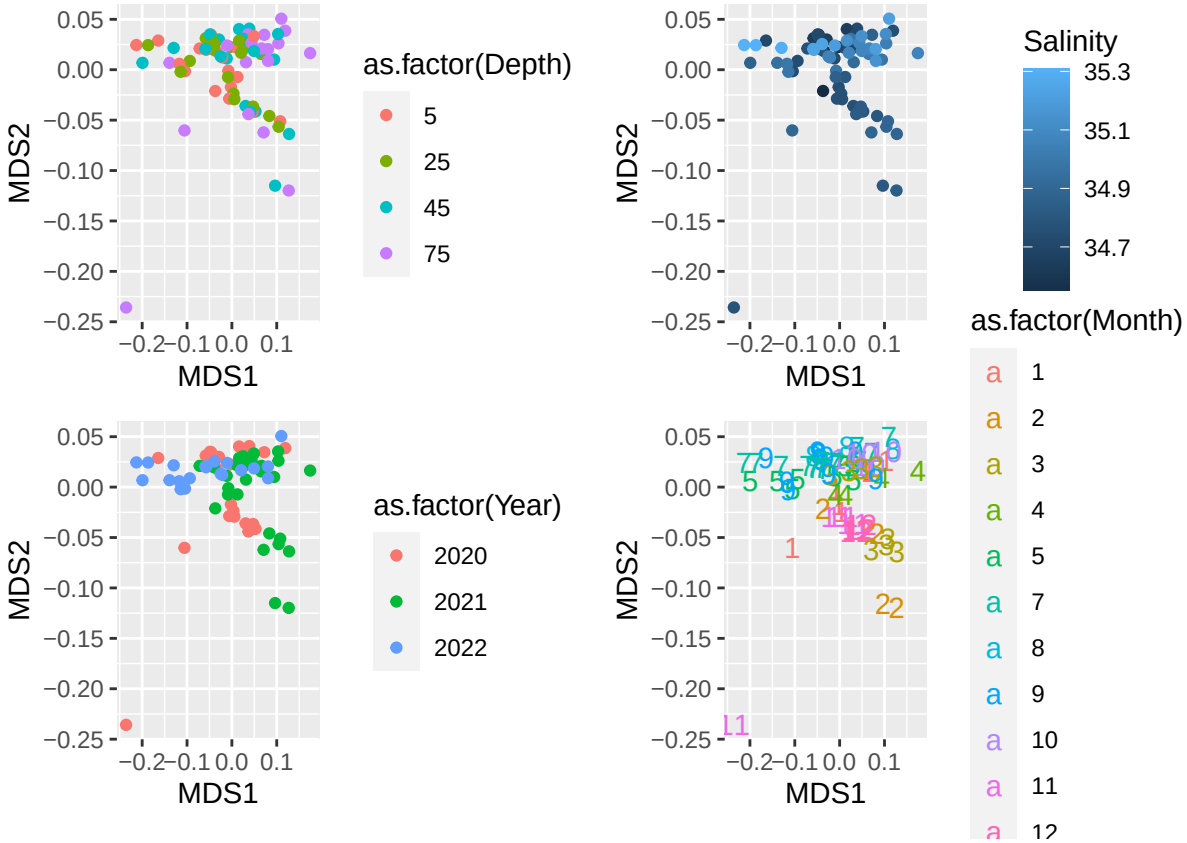
```
## [1] 76 16
```

```
shallow.rel.abundance <- decostand(shallow.data[,categories], method="total", MARGIN = 1)
shallow.distance.matrix <- as.matrix( vegdist(shallow.rel.abundance, method="chisq"),
                                      labels = T)
nMDS.shallow <- metaMDS(shallow.distance.matrix, distance = "chisq", k = 2, trace = 0)
nMDS.shallow$stress
```

```
## [1] 0.02374406
```

```
shallow.data$MDS1 <- nMDS.shallow$points[,"MDS1"]
shallow.data$MDS2 <- nMDS.shallow$points[,"MDS2"]
```

```
p1 <- ggplot(data = shallow.data, aes(x = MDS1, y = MDS2, col = as.factor(Depth))) +
  geom_point()
p2 <- ggplot(data = shallow.data, aes(x = MDS1, y = MDS2, col = Salinity)) +
  geom_point()
p3 <- ggplot(data = shallow.data, aes(x = MDS1, y = MDS2, col = as.factor(Year))) +
  geom_point()
p4 <- ggplot(data = shallow.data, aes(x = MDS1, y = MDS2, col = as.factor(Month))) +
  geom_text(data = shallow.data, aes(label = Month))
p1 + p2 + p3 + p4 + plot_layout(ncol = 2)
```



This last attempt doesn't so much affect the previous finding, but it shows a property of distance and dissimilarity metrics. Since chi-squared distance satisfies the mathematical definition of a distance, running nMDS on a subset of points produces a cluster whose arrangement is similar to the arrangement of these points within the larger set.