Introduction to non-metric multidimensional scaling

Getting started

Load packages

Let's load the packages.

```
library(MASS)
library(vegan)
#library(labdsv)
library(tidyverse)
theme_set(theme_bw(base_size = 14))

## Define the repository from which we get the data
repo_url <- "https://raw.githubusercontent.com/elahi/phd_elahi/master/northeast_pacific_wall_surveys/data/"
```

The data

```
dat <- read.csv(paste(repo_url, "sjc_sessile_130513_mds.csv", sep = ""))
glimpse(dat)
```

```
## Observations: 864
## Variables: 80
## $ orig.order <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1...
## $ quad       <fct> ON0_45, ON0_46, ON0_48, ON0_49, ON1_17, ON1_84, ON1_85, ...
## $ transect   <fct> ON0, ON0, ON0, ON0, ON1, ON1, ON1, ON1, ON2, ON2, ON2, O...
## $ site       <fct> ON, ON, ON, ON, ON, ON, ON, ON, ON, ON, ON, ON, ON, ON, ...
## $ real.date  <int> 39424, 39424, 39424, 39424, 39424, 39424, 39424, 39424, ...
## $ comm.date  <int> 39426, 39426, 39426, 39426, 39426, 39426, 39426, 39426, ...
## $ month      <int> 712, 712, 712, 712, 712, 712, 712, 712, 712, 712, 712, 7...
## $ urch       <fct> u_con, u_con, u_con, u_con, u_rem, u_rem, u_rem, u_rem, ...
## $ chit       <fct> c_con, c_rem, c_rem, c_con, c_rem, c_con, c_rem, c_con, ...
## $ treat      <fct> u_con.c_con, u_con.c_rem, u_con.c_rem, u_con.c_con, u_re...
## $ treat.ch   <int> 19, 2, 2, 19, 1, 8, 1, 8, 2, 2, 19, 19, 8, 8, 1, 1, 2, 2...
## $ BASP       <dbl> 0.079606649, 0.190481341, 0.172054104, 0.061217549, 0.01...
## $ CCAP       <dbl> 0.035354718, 0.041103299, 0.105456213, 0.034009749, 0.00...
## $ EALM       <dbl> 0.438070709, 0.383017847, 0.295735901, 0.408116994, 0.76...
## $ RFSP       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ OPCA       <dbl> 0.0000000, 0.0000000, 0.0000000, 0.0000000, 0.0000000, 0...
## $ RBSP       <dbl> 0.004682744, 0.004326663, 0.006877579, 0.000000000, 0.00...
## $ AGFI       <dbl> 0.0000000, 0.0000000, 0.0000000, 0.0000000, 0.0000000, 0...
## $ BFSP       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ DICL       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ Cris       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ HEPA       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ PHPA       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ BRUP       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ HBCO       <dbl> 0.035120581, 0.006489995, 0.151306740, 0.015871216, 0.00...
## $ SPS1       <dbl> 0.032779209, 0.151433207, 0.112448418, 0.034009749, 0.00...
```

```
## $ SPS3       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ SPS8       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ SPS9       <dbl> 0.000000000, 0.000000000, 0.006877579, 0.000000000, 0.00...
## $ SPS10      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ SPOT       <dbl> 0.000000000, 0.000000000, 0.004585053, 0.002267317, 0.00...
## $ MSP1       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ SYSP       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ SCUN       <dbl> 0.002458441, 0.000000000, 0.006877579, 0.000000000, 0.00...
## $ BRSH       <dbl> 0.014048232, 0.012979989, 0.022925264, 0.024940483, 0.03...
## $ EUBI       <dbl> 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.000000...
## $ DELI       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ HYSP       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ EPSC       <dbl> 0.0000000, 0.0000000, 0.0000000, 0.0000000, 0.0000000, 0...
## $ BAEL       <dbl> 0.000000000, 0.004326663, 0.002407153, 0.022673166, 0.00...
## $ MEFA.t     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ GERU       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ DOCO       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ DOFE       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ TUBE       <dbl> 0.051744322, 0.002163332, 0.006877579, 0.000000000, 0.00...
## $ PSOC       <dbl> 0.002926715, 0.006598161, 0.002407153, 0.018138533, 0.00...
## $ VECO       <dbl> 0.007024116, 0.019469984, 0.011462632, 0.000000000, 0.00...
## $ BACR       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ PSCH       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ TETR       <dbl> 0.000117069, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ CHHA       <dbl> 0.002341372, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ POMA       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ META       <dbl> 0.084991805, 0.028231476, 0.000000000, 0.034123115, 0.03...
## $ PYST       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ PEAN       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CLHU       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ DICA       <dbl> 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.000000...
## $ TCM3       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ DITR       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ EUPU       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ SYPA       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ APPO       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ APSO       <dbl> 0.000000000, 0.004326663, 0.000000000, 0.000000000, 0.00...
## $ TCM2       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ TCOT       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ BOVI       <dbl> 0.000117069, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ Chel       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ Core       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CNFI       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ HAIG       <dbl> 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.000000...
## $ Stye       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ ASPA       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ TSV1       <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.00...
## $ TSV2       <dbl> 0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.000000...
## $ TSOT       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ INOT       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ OTHE       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ SHEL       <dbl> 0.098454694, 0.129908058, 0.036680422, 0.335562861, 0.01...
## $ FUZZ       <dbl> 0.070358230, 0.015143321, 0.055020633, 0.009069267, 0.10...
## $ RUBB       <dbl> 0.039803325, 0.000000000, 0.000000000, 0.000000000, 0.00...
```

```
dim(dat)
summary(dat)
```

```r
colnames(dat)
str(dat)

## subset 1009 data (to best match NEPWS sampling)
dat1 <- subset(dat, dat$month=="1009")
dim(dat1)
colnames(dat1)

# create spp matrix (including ON and PG)
abun <- dat1[,13:75]
summary(abun)

# transform matrix
abun.sq <- sqrt(abun)

# presence/absence
abun.pa <- decostand(abun, method="pa")

# create group matrix
group <- dat1[,1:12]

# Perform the NMDS in 2 dimensions on raw data
abun.MDS2 = metaMDS(abun, distance="bray", k=2, engine = 'monoMDS', autotransform=FALSE, noshare=0.1, tryma
abun.MDS2$stress # stress = 0.225

# Perform the NMDS in 2 dimensions on sqrt data
abun.sq.MDS2 = metaMDS(abun.sq, distance="bray", k=2, engine = 'monoMDS', autotransform=FALSE, noshare=0.1,
abun.sq.MDS2$stress # stress = 0.236


# Perform the NMDS in 2 dimensions on sqrt data
abun.pa.MDS2 = metaMDS(abun.pa, distance="bray", k=2, engine = 'monoMDS', autotransform=FALSE, noshare=0.1,
abun.pa.MDS2$stress # stress = 0.241



# plot 3 panels
par(mfrow=c(1,3), pty="s", mar=c(1,1,1,1), oma=c(2,4,2,1))

plot(abun.MDS2$points, col=c(group$site), xaxt='n', yaxt='n', xlab='', ylab='', main="Raw cover")
plot(abun.sq.MDS2$points, col=c(group$site), xaxt='n', yaxt='n', xlab='', ylab='', main="Square root")
plot(abun.pa.MDS2$points, col=c(group$site), xaxt='n', yaxt='n', xlab='', ylab='', main="Presence-Absence")

legend("topright", legend=levels(group$site), pch=1, col=c(1:3), bty='y')


# now do the same for the NEPWS data (pa only)
dat2 <- read.csv(paste(repo_url, "NEPWS_channel_130513.csv", sep = ""), header=TRUE, na.strings="NA")
colnames(dat2)

# create spp matrix (including ON and PG)
abun2 <- dat2[,15:87]
summary(abun2)

# create group matrix
group2 <- dat2[,1:14]
```

```r
group2

# Perform the NMDS in 2 dimensions on raw data
nepws.MDS2 = metaMDS(abun2, distance="bray", k=2, engine = 'monoMDS', autotransform=FALSE, noshare=0.1, try
nepws.MDS2$stress # stress = 0.264

# compare both PA plots

par(mfrow=c(1,2), pty="s", mar=c(1,1,1,1), oma=c(2,4,2,1))

plot(nepws.MDS2$points, col=c(group$site), xaxt='n', yaxt='n', xlab='', ylab='', main="NEPWS")
legend("topright", legend=levels(group2$site), pch=1, col=c(1:3), bty='y')


plot(abun.pa.MDS2$points, col=c(group$site), xaxt='n', yaxt='n', xlab='', ylab='', main="Presence-Absence_%
```

More Practice

7. Mutate the data frame so that it includes a new variable that contains the average speed, `avg_speed` traveled by the plane for each flight (in mph). **Hint:** Average speed can be calculated as distance divided by number of hours of travel, and note that `air_time` is given in minutes. Make a scatterplot of `avg_speed` vs. `distance`. **Hint:** Use `geom_point()`. Describe the relationship between average speed and distance.
8. Replicate the following plot. **Hint:** The data frame plotted only contains flights from American Airlines, Delta Airlines, and United Airlines, and the points are `colored` by `carrier`. Include the code you used to create the plot, as well as the plot itself.

License