Ordination 2

Lecture 10.1 Permutational Anova

Lauren Sullivan

Module: Multivariate Models

Readings

Required for class:

- NA

Optional:

- Anderson, M. J. (2001) A new method for non-parametric multivariate analysis of variance. *Austral Ecology*.
- McArdle, B. H. and Anderson, M. J. (2001) Fitting multivariate models to community data: A comment on distance-based redundancy analysis. *Ecology*.

Multivariate Analysis

There are several ways to look at multivariate patterns from a matrix of **Y**'s.

1. Linear models: MANOVA/regression to test patterns
2. Ordination: PCA, nMDS, etc to visualize patterns
3. **Permutation tests: PERMANOVA to test patterns**

Revisiting our Zooplankton

How does increasing connectivity of waterways alter aquatic zooplankton community?

Journal of Applied Ecology

BRITISH ECOLOGICAL SOCIETY

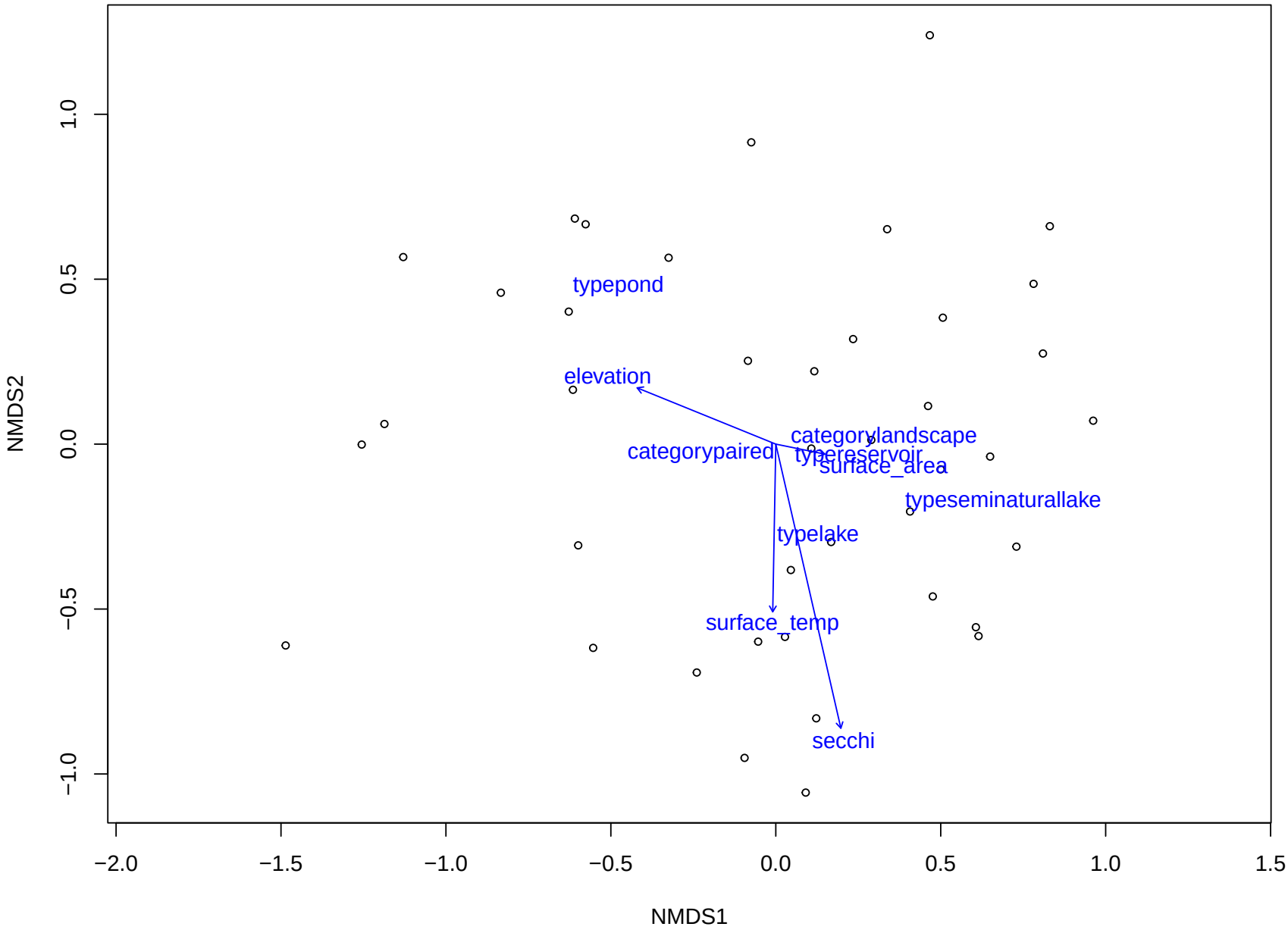
Journal of Applied Ecology 2017, **54**, 1343–1352 doi: 10.1111/1365-2664.12882

Increased habitat connectivity homogenizes freshwater communities: historical and landscape perspectives

Angela L. Strecker* and Jeffrey T. Brittain

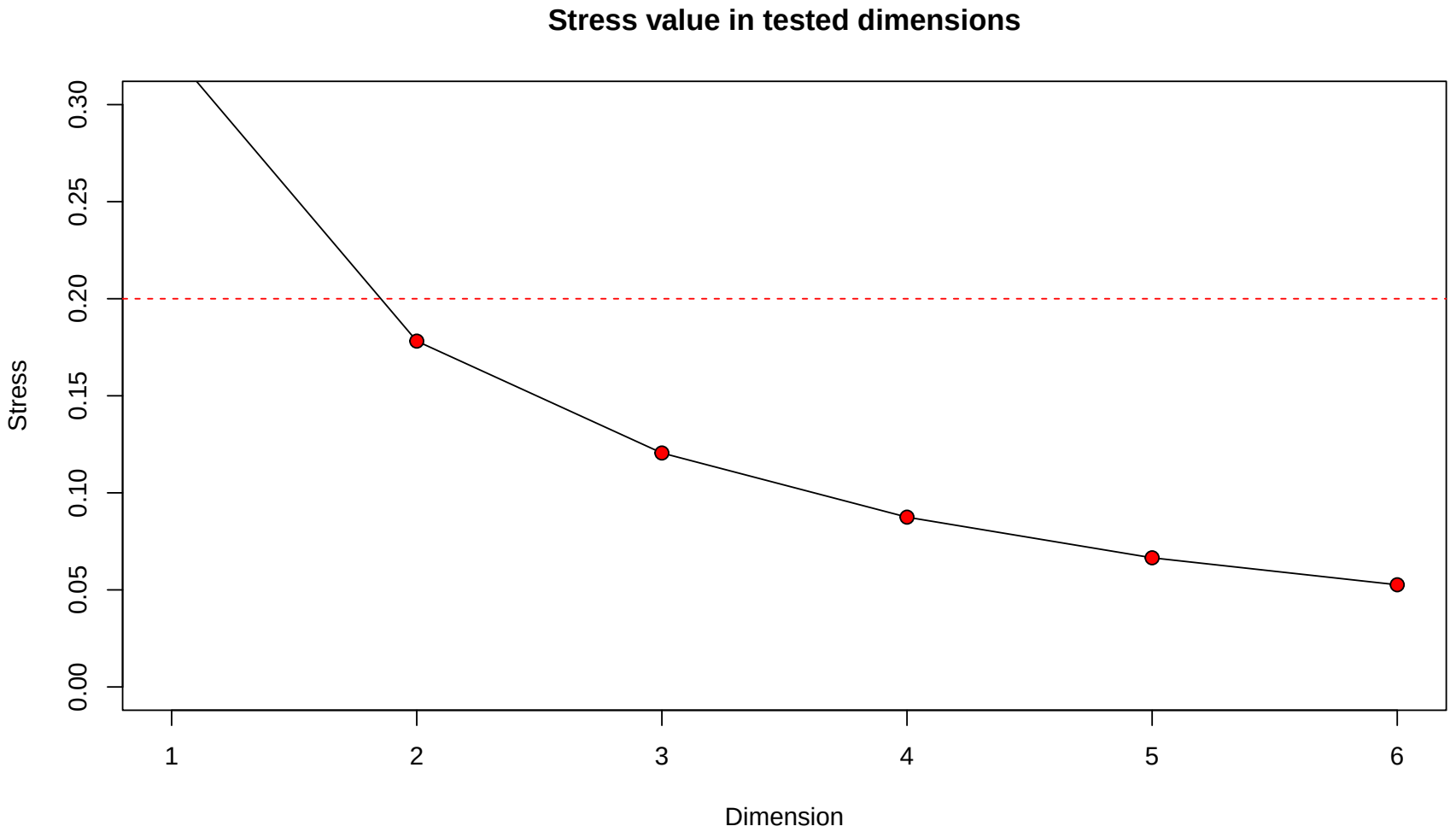
Department of Environmental Science and Management, Portland State University, Portland, OR, USA

nMDS figure of Zooplankton Communities



Dimension Checks

```
dimcheckMDS(dat[,-c(1, 43:48)], distance = "bray", k = 6, trymax = 20,
            autotransform = FALSE)
```

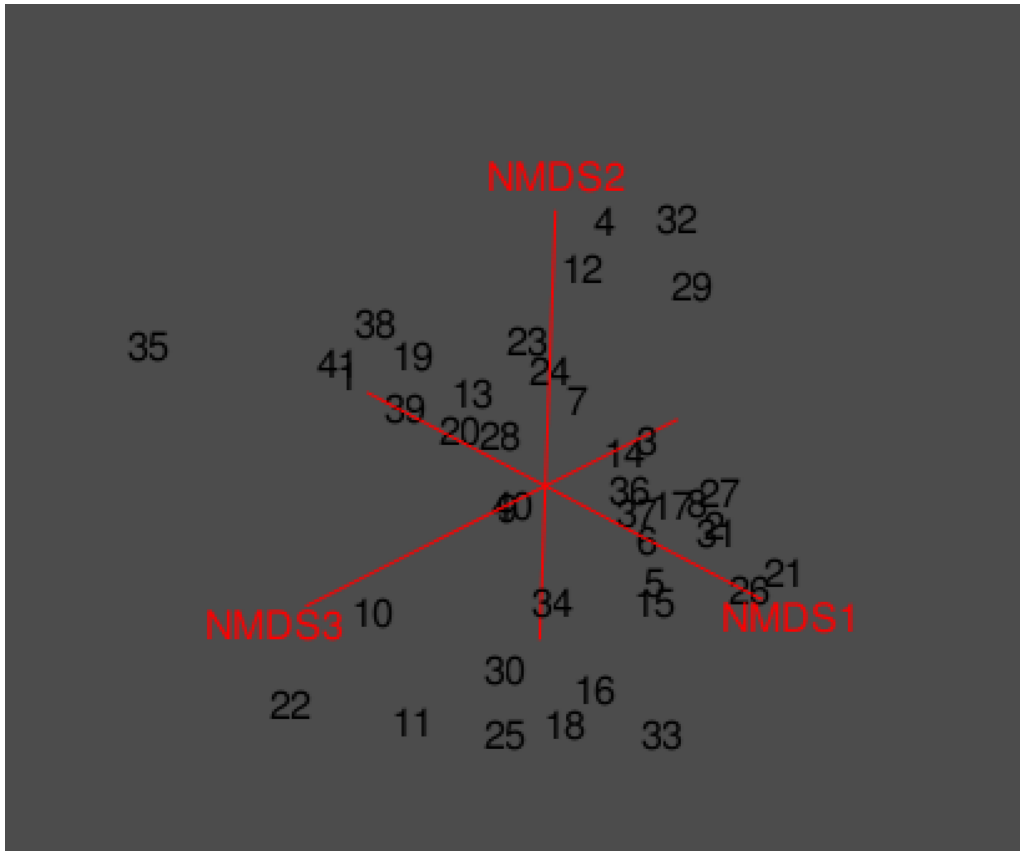



```
#in library("goeveg")
```

3D nMDS

You can also plot 3D nMDS plots with `library(vegan3d)`. You also need to have `library(rgl)` installed. This figure is interactive!

```
dat.mds3 <- metaMDS(dat[,-c(1, 43:48)], k = 3, distance = 'bray',
                    autotransform = F, expand = F)
ordirgl(dat.mds3, type = "t")
```



Relationships between X's and Sites Patterns.

How to determine if there is a relationship between the plotted environmental variables, and site-level community patterns?

1. Principle Components Analysis (PCA) / Principle Coordinates Analysis (PCoA)
2. Canonical Correspondence Analysis (CCA) / Redundancy Analysis (RDA)

- For both, relationships in predictors are also forced to be linear. These are typically non-distance based (but see db-RDA).

3. Permutational ANOVA (PERMANOVA)

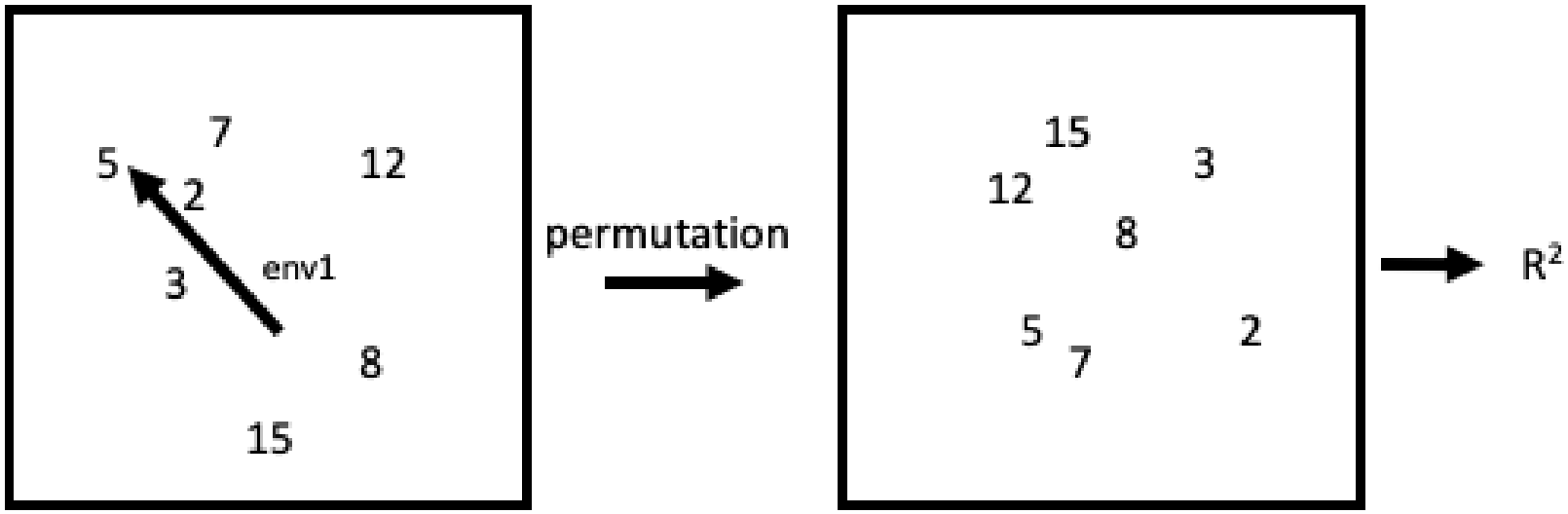
- Are the relative lengths of the arrows that represent environmental variables in your nMDS plot predictive?

Permutational ANOVA (PERMANOVA)

Given a set of site scores (Axis 1, Axis 2, ...) and environmental information at each site, this procedure calculates R^2 for Site Scores ~ Environment for each iteration.

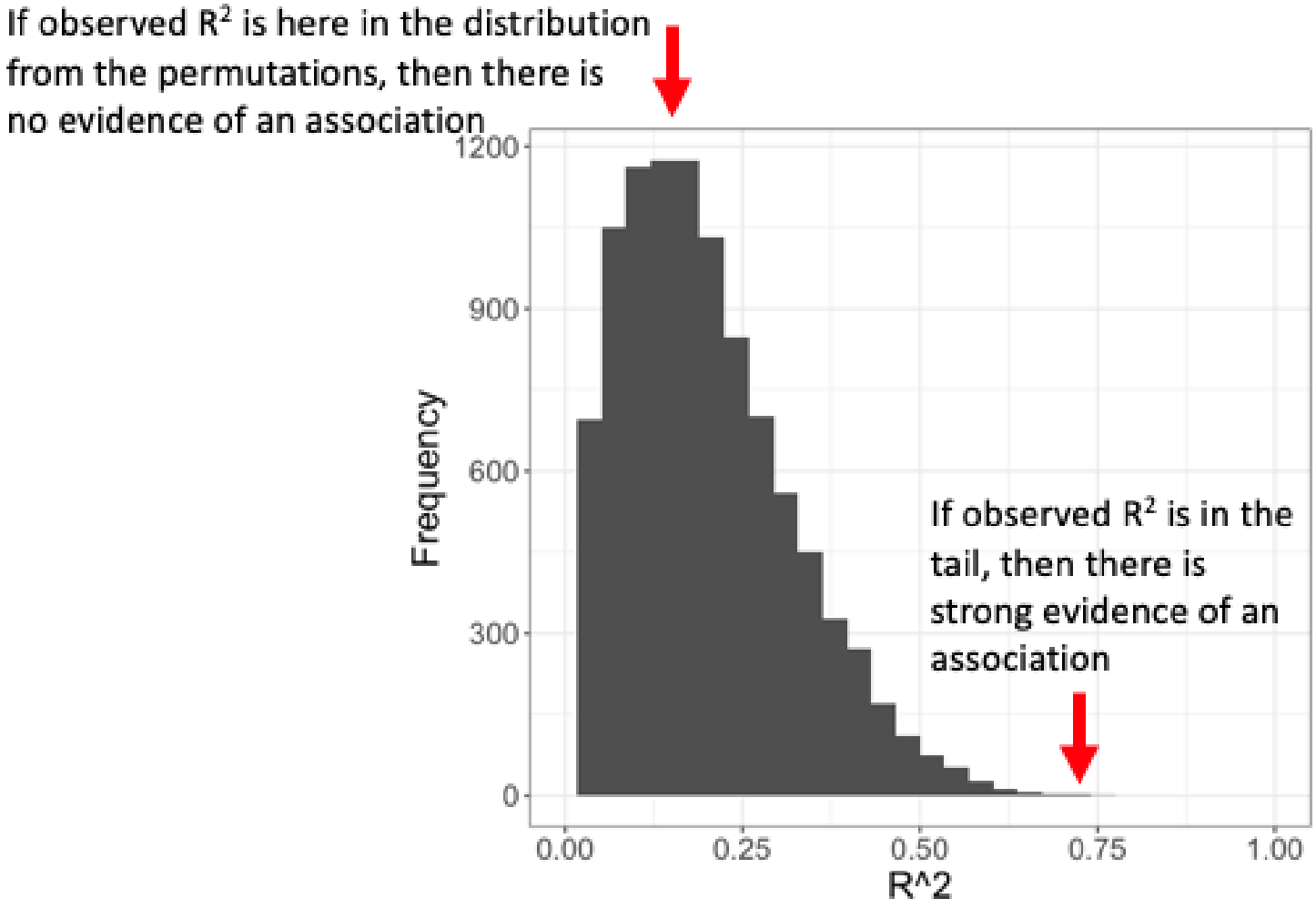
- H_0 = no correlation (correlation = 0)

PERMANOVA holds the Y data constant and shuffles the X's (environemntal data/trt labels)



Permutational ANOVA (PERMANOVA)

The relative location of the observed R^2 to the permuted R^2's indicates association strength.



PERMANOVA with `adonis()`.

Let's examine: community pattern ~ secchi depth + site type.

- This analysis deals with X's sequentially, so order of terms matters unless data is balanced.

```
dat.bray <- vegdist(zoop[,-1], "bray")
adonis(dat.bray ~ secchi + type, data = dat.env )
```

```
## $aov.tab
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## secchi     1    1.5491 1.54914  5.0152 0.10198  0.001 ***
## type       4    2.8300 0.70751  2.2905 0.18631  0.001 ***
## Residuals 35   10.8110 0.30889         0.71171
## Total     40   15.1902                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $call
## adonis(formula = dat.bray ~ secchi + type, data = dat.env)
##
## $coefficients
```

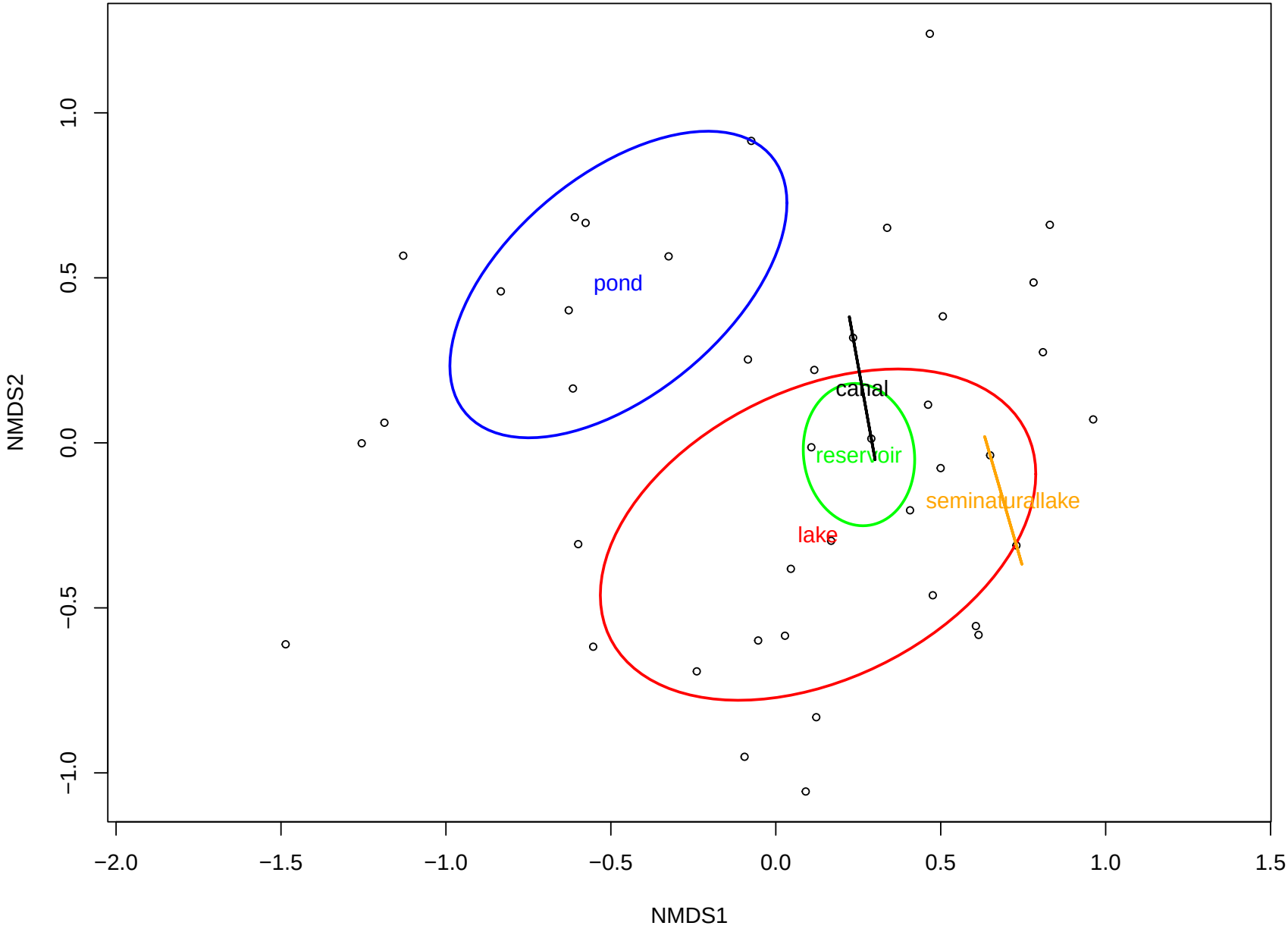
Blocking in `adonis()`

You can include block effects in `adonis()` with `strata=`. Here, permutations happen within these groups, or blocks.

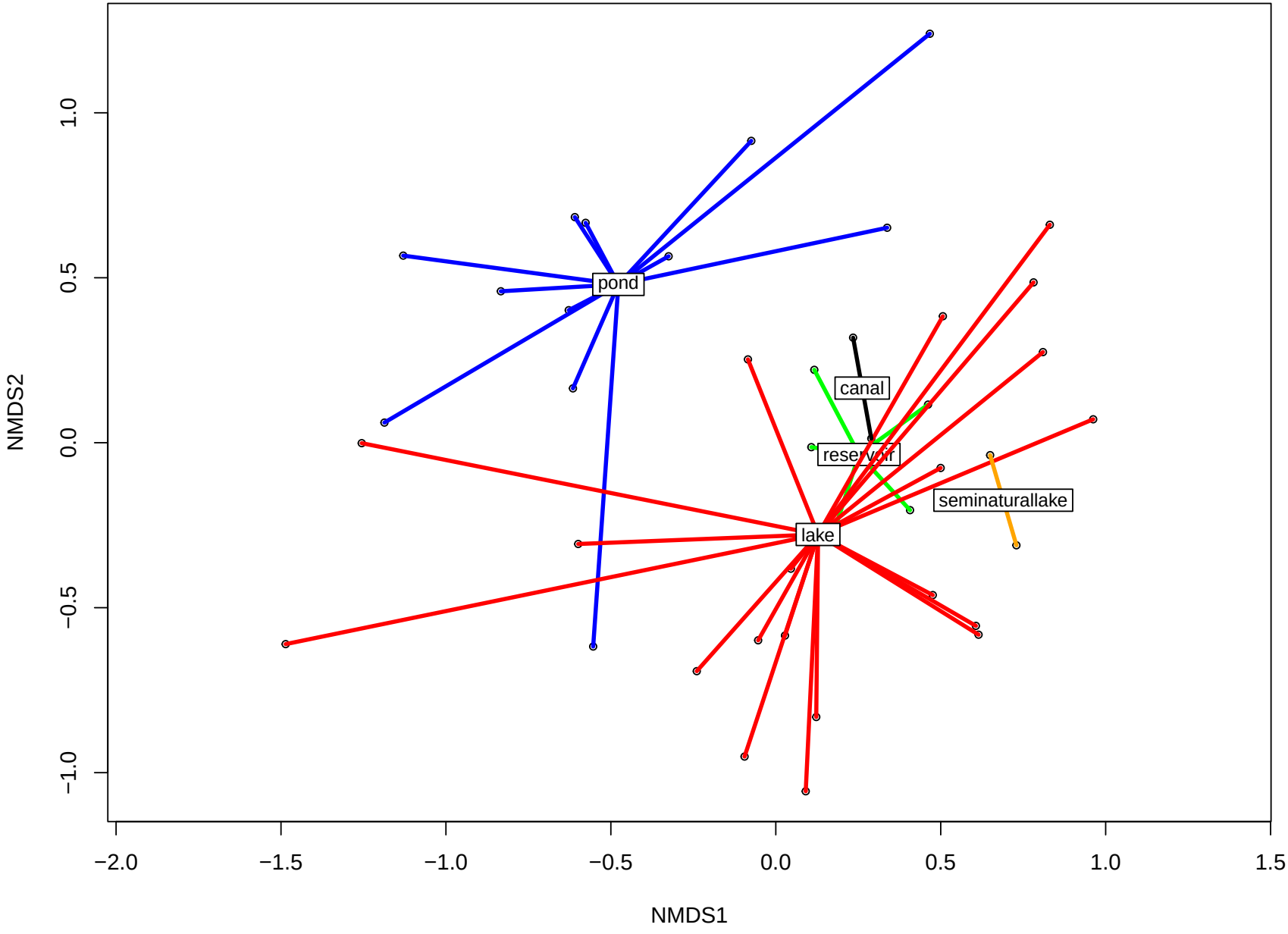
```
adonis(dat.bray ~ secchi, data = dat.env, strata = dat.env$type )
```

```
## $aov.tab
## Blocks:  strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## secchi     1    1.5491 1.54914   4.429 0.10198  0.013 *
## Residuals 39   13.6410 0.34977         0.89802
## Total     40   15.1902                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $call
## adonis(formula = dat.bray ~ secchi, data = dat.env, strata = dat.env$type)
##
## $coefficients
## NULL
##
## $coef.sites
```

Prettier nMDS figures



Prettier nMDS figures



Dispersion Within Groups

```
betadisper(dat.bray, dat.env$type)
```

```
##
##  Homogeneity of multivariate dispersions
##
## Call: betadisper(d = dat.bray, group = dat.env$type)
##
## No. of Positive Eigenvalues: 27
## No. of Negative Eigenvalues: 13
##
## Average distance to median:
##           canal            lake            pond       reservoir seminaturallake
##          0.2267          0.5684          0.5708          0.3341          0.2754
##
## Eigenvalues for PCoA axes:
## (Showing 8 of 40 eigenvalues)
##  PCoA1  PCoA2  PCoA3  PCoA4  PCoA5  PCoA6  PCoA7  PCoA8
## 3.0771 2.6221 1.8629 1.5036 1.1531 0.9901 0.9037 0.6042
```

Extracting nMDS Scores

You can extract the nMDS scores for the number of dimensions you choose.

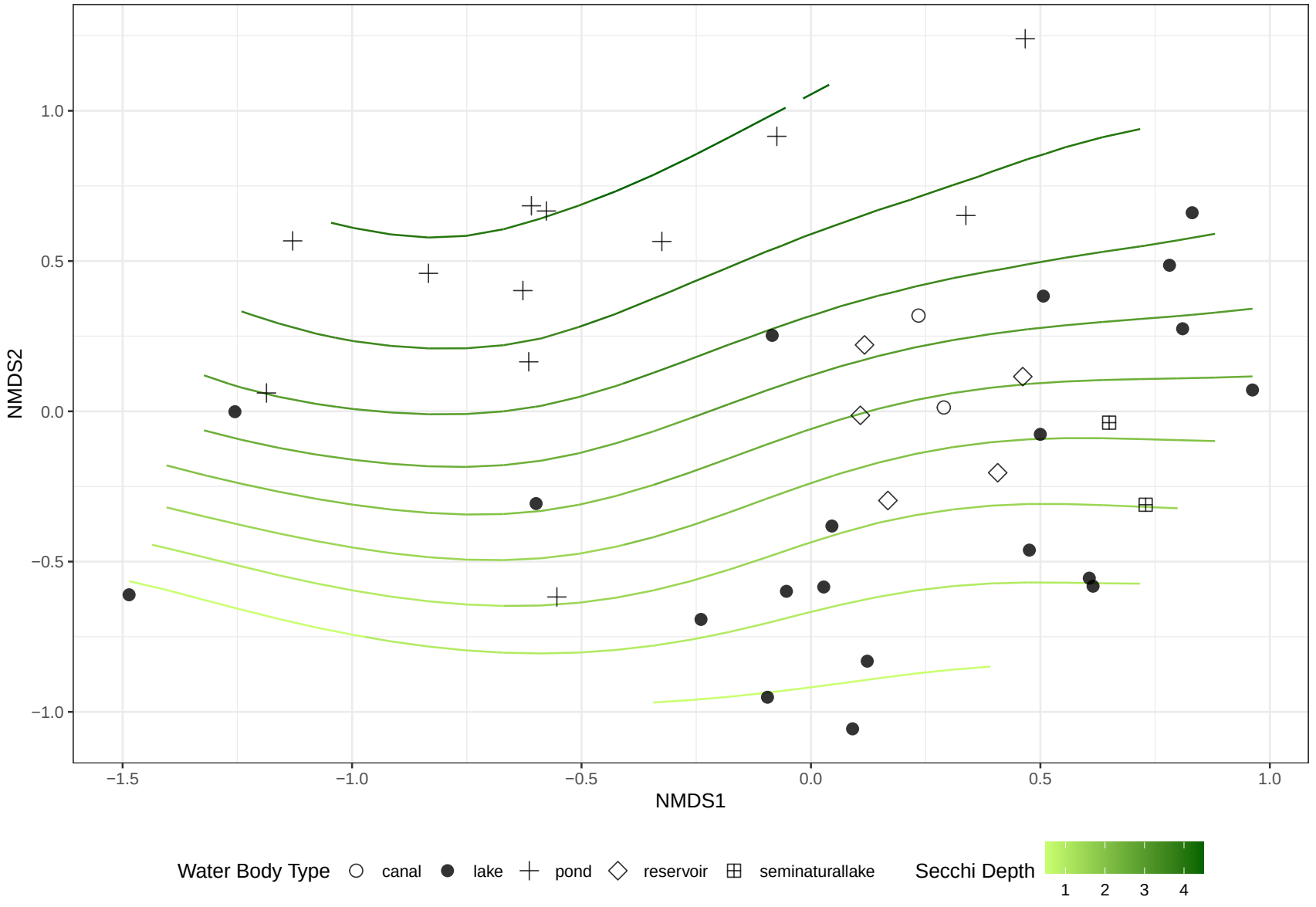
```
head(dat.mds2$points)
```

```
##          MDS1        MDS2
## 1 -1.18645504  0.06104100
## 2  0.40696695 -0.20432815
## 3  0.16769391 -0.29679407
## 4 -0.09470387 -0.95121687
## 5  0.23434768  0.31847649
## 6  0.28953200  0.01271891
```

```
head(dat.mds3$points)
```

```
##          MDS1        MDS2        MDS3
## 1 -1.3851208  0.19691751  0.16823711
## 2  0.7124502  0.02208257 -0.26309983
## 3  0.3644399  0.29473895 -0.22828203
## 4  0.2632652  1.20067073  0.03108167
## 5  0.3525979 -0.58457064 -0.44357955
## 6  0.4105909 -0.19321662 -0.24204672
```

Plotting with `ggplot()`



A Few nMDS and PERMANOVA Notes

- Set a seed if you want your results to be 100% reproducible.
  - Make sure your results are not dependant on your seed choice.
  - To do this, you just make sure you type `set.seed(value)` before you run the nMDS.
- PERMANOVA does better with a larger sample size.
- PERMANOVA works best with balanced data that's homoscedastic.

Metabolomics example

Let's run through an example looking at metabolite data for different accessions of plants. The data is already standardized, so no need to worry about that!

```
## # A tibble: 40 x 6
##    access    unknown_A1 unknown_1 amide_1 amide_12  amide_2
##    <chr>          <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
##  1 Ang267             0     0     0.00754   0.0689 0.0314
##  2 AngAng272          0     0     0.0192    0.0666 0.0568
##  3 AngAng285          0     0     0.0292    0.212  0.0749
##  4 AngAng318          0     0.168 0         0.219  0.0314
##  5 AngStr266          0     0     0.0340    0.267  0.0802
##  6 AngStr320          0     0     0.00425   0      0.133
##  7 Atr255             0     0     0.275     0.312  0.00832
##  8 Atr260             0     0     0.148     0.355  0.164
##  9 Atr262             0     0     0.0638    0.161  0.000876
## 10 Atr299             0     0     0.0707    0.286  0.00179
## # ... with 30 more rows
```

Metabolomics example

One thing we can do with this data that we have not done yet is look at dissimilarity within and between groups.

```
#add species names
lipo$spp <-substring(lipo$access, 1, 3)
lipo[1:10, c(1,45)]
```

```
## # A tibble: 10 x 2
##    access    spp
##    <chr>     <chr>
##  1 Ang267    Ang
##  2 AngAng272 Ang
##  3 AngAng285 Ang
##  4 AngAng318 Ang
##  5 AngStr266 Ang
##  6 AngStr320 Ang
##  7 Atr255    Atr
##  8 Atr260    Atr
##  9 Atr262    Atr
## 10 Atr299    Atr
```

Metabolomics example

Calculate within group and between group dissimilarity

```
#determine which species are the same and which are not
temp<-outer(lipo$spp, lipo$spp, '!=')

#as.dist computes a distance matrix between the TRUE and FALSE objects
#   calculated above
lipo.DN <- as.dist(temp)
lipo.DN
```

```
##    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 2
## 2  0
## 3  0 0
## 4  0 0 0
## 5  0 0 0 0
## 6  0 0 0 0 0
## 7  1 1 1 1 1 1
## 8  1 1 1 1 1 1 0
## 9  1 1 1 1 1 1 0 0
## 10 1 1 1 1 1 1 0 0 0
## 11 1 1 1 1 1 1 1 1 1  1
## 12 1 1 1 1 1 1 1 1 1  1  0
## 13 1 1 1 1 1 1 1 1 1  1  1  1
## 14 1 1 1 1 1 1 1 1 1  1  1  1  0
## 15 1 1 1 1 1 1 1 1 1  1  1  1  0  0
```

Metabolomics example

Create your distance matrix using Canberra distance which is common for metabolomic work.

```
######Calculate Canberra Distance matrix
lipo.tot <- decostand(lipo[,-c(1,45)], "total")
lipo.CB<-vegdist(lipo.tot, method="canberra")
lipo.CB
```

```
##           1         2         3         4         5         6         7
## 2  0.3997161
## 3  0.5566720 0.3549826
## 4  0.6685866 0.5463446 0.4678559
## 5  0.3865719 0.4162030 0.4326886 0.6046813
## 6  0.8492538 0.8815037 0.9373899 0.9662237 0.8835482
## 7  0.5368665 0.5996072 0.6427640 0.6755673 0.5854728 0.9182863
## 8  0.4940075 0.5087128 0.5609677 0.6770674 0.5237182 0.9135545 0.3665594
## 9  0.5521353 0.5671033 0.5776004 0.6323172 0.5989294 0.9575724 0.3773248
## 10 0.4951283 0.5596591 0.5588894 0.6230645 0.5491866 0.9322980 0.3988799
## 11 0.9324943 0.9349387 0.9144767 0.9107048 0.9435805 0.4636249 0.9495876
## 12 0.9024994 0.9108419 0.8996113 0.9032825 0.9261152 0.5412162 0.9423477
## 13 0.8040965 0.8008824 0.7953170 0.8284799 0.7936918 0.7723194 0.8138786
## 14 0.8032811 0.7819685 0.8038406 0.8410878 0.7866245 0.7539185 0.7855628
## 15 0.8430317 0.8298188 0.8193983 0.8266771 0.8141567 0.8405302 0.7897776
## 16 0.8070849 0.8017122 0.7978571 0.8268532 0.7860922 0.7901156 0.7639194
## 17 0.6896286 0.7254036 0.6589608 0.7688955 0.6580339 0.8561196 0.6102925
```

Metabolomics example

You can calculate the average dissimilarity between groups (within species, and between species)

```
mean(lipo.CB[lipo.DN==0])
```

```
## [1] 0.4630613
```

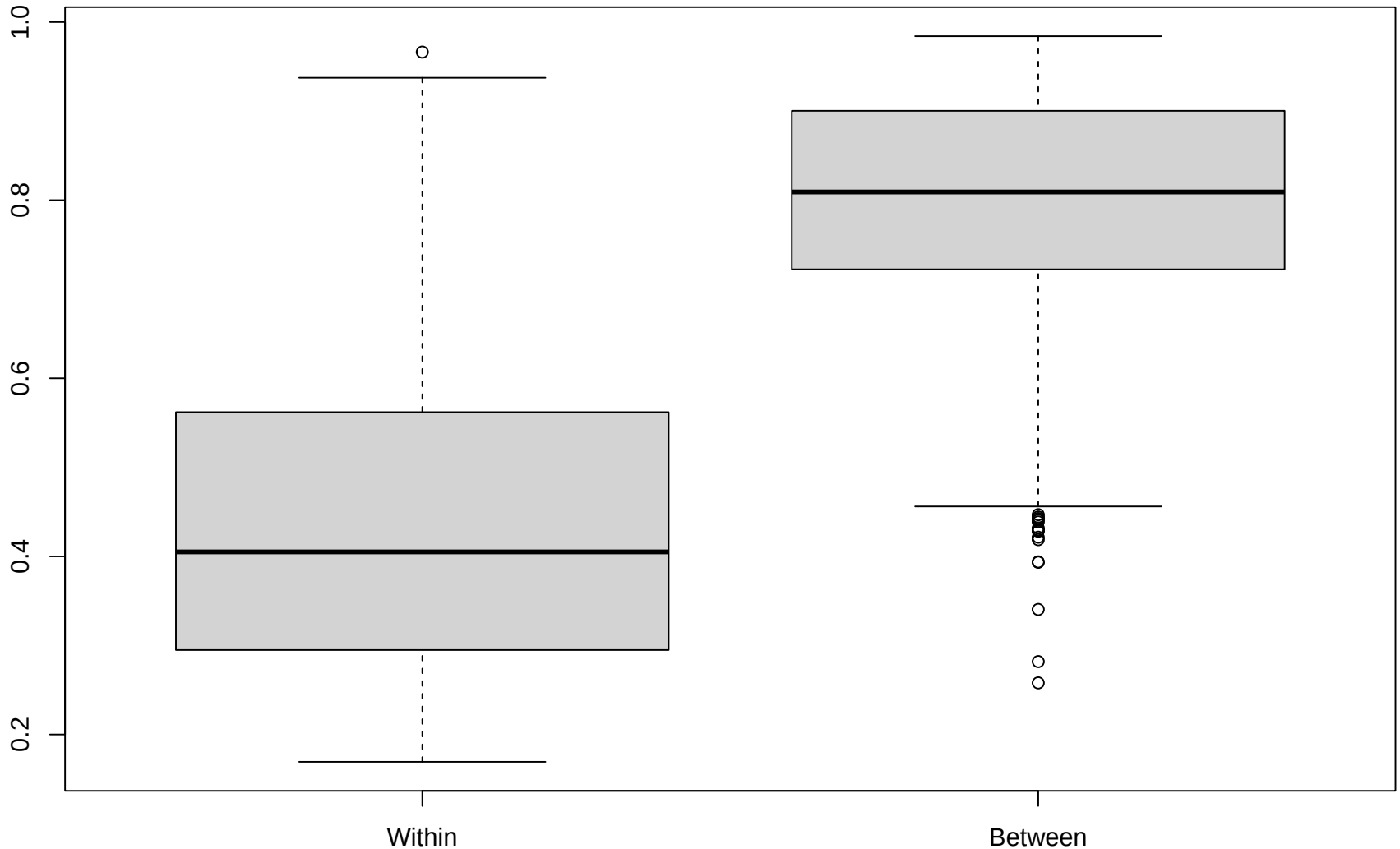
```
mean(lipo.CB[lipo.DN==1])
```

```
## [1] 0.7876432
```

Metabolomics example

Or simply plot the dissimilarity within and among species.

```
boxplot(split(lipo.CB, lipo.DN), names=c('Within','Between'))
```



Metabolomics example

Let's run a permutational ANOVA to determine if there is a difference in metabolomic structure among species.

```
lipo.adonis <-adonis(lipo.CB ~ spp, data=lipo)
lipo.adonis
```

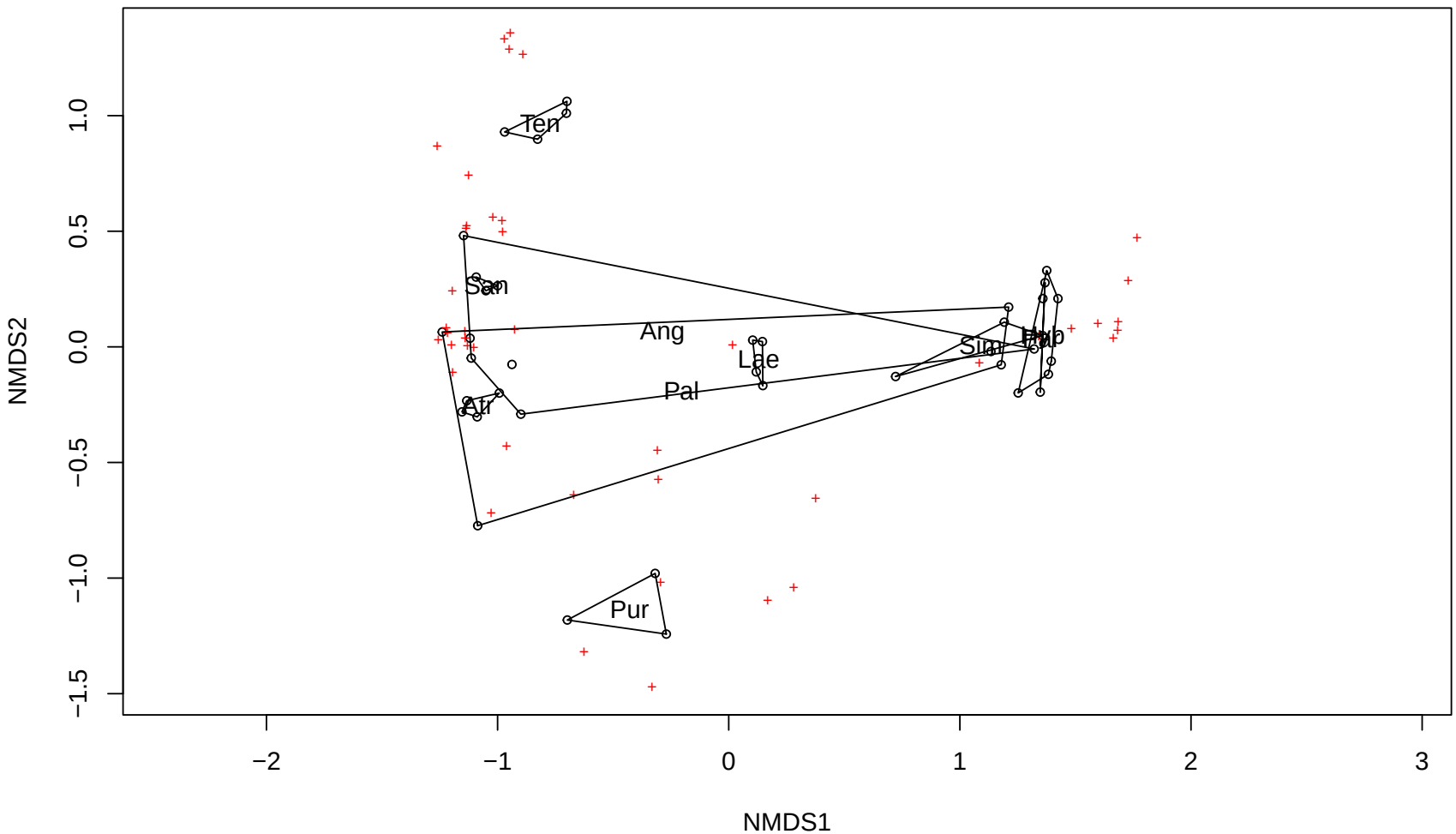
```
## $aov.tab
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## spp        9    8.2845 0.92050   7.795 0.70046  0.001 ***
## Residuals 30    3.5426 0.11809         0.29954
## Total     39   11.8271                1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $call
## adonis(formula = lipo.CB ~ spp, data = lipo)
##
## $coefficients
## NULL
##
```

Metabolomics example

And plot those results

```
lipo.mds2 <- metaMDS(lipo[,-c(1,45)],k=2,distance='canberra',autotransform=F)
```

```
plot(lipo.mds2)
ordihull(lipo.mds2, lipo$spp, label = T)
```



Metabolomics example

So it looks like there might be a difference in the dispersion between species Let's test this. First calculate your dispersion within species with `betadisper()`.

```
lipo.bd <-betadisper(lipo.CB, lipo$spp)
lipo.bd
```

```
##
##  Homogeneity of multivariate dispersions
##
## Call: betadisper(d = lipo.CB, group = lipo$spp)
##
## No. of Positive Eigenvalues: 32
## No. of Negative Eigenvalues: 7
##
## Average distance to median:
##    Ang    Atr    Hyb    Lae    Pal    Par    Pur    San    Sim    Ten
## 0.3900 0.2092 0.1343 0.1676 0.4194 0.2806 0.2592 0.1082 0.2366 0.1972
##
## Eigenvalues for PCoA axes:
## (Showing 8 of 39 eigenvalues)
##  PCoA1  PCoA2  PCoA3  PCoA4  PCoA5  PCoA6  PCoA7  PCoA8
## 4.2158 1.5342 1.1916 0.7808 0.7049 0.5653 0.4789 0.4023
```

Metabolomics example

Then compare differences in dispersal between species overall with `anova()`, and using pairwise comparisons with `TukeyHSD()`.

```
anova(lipo.bd)
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value  Pr(>F)
## Groups     9 0.39339 0.043709  2.0071 0.07388 .
## Residuals 30 0.65334 0.021778
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We fail to reject the null hypothesis that dispersion is equal across all species

```
TukeyHSD(lipo.bd)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                diff        lwr        upr     p adj
## Atr-Ang -0.18076901 -0.5057123 0.14417430 0.6704801
```