

Strata

- The strata option allows you to control how permutations are done. Specifically, to constrain permutations.
- Why would you want to do this?
- In this dataset, there are clear differences in “area” (A vs. B), but what if our main interest was “treatment” (T1 vs. T2)

	area	treatment	sp1	sp2	sp3	sp4	sp5	sp6
sam1	A	T1	35	32	15	2	10	4
sam2	A	T2	27	20	14	4	13	2
sam3	A	T1	29	34	26	3	20	5
sam4	A	T2	17	26	22	5	11	4
sam5	A	T1	32	29	24	3	16	4
sam6	A	T2	16	19	21	6	9	3
sam7	B	T1	5	4	22	27	20	18
sam8	B	T2	1	2	28	18	14	12
sam9	B	T1	6	4	15	39	19	14
sam10	B	T2	5	2	20	41	16	16
sam11	B	T1	3	5	25	29	19	17
sam12	B	T2	2	1	18	16	15	13

Strata

	area	treatment	sp1	sp2	sp3	sp4	sp5	sp6
sam1	A	T1	35	32	15	2	10	4
sam2	A	T2	27	20	14	4	13	2
sam3	A	T1	29	34	26	3	20	5
sam4	A	T2	17	26	22	5	11	4
sam5	A	T1	32	29	24	3	16	4
sam6	A	T2	16	19	21	6	9	3
sam7	B	T1	5	4	22	27	20	18
sam8	B	T2	1	2	28	18	14	12
sam9	B	T1	6	4	15	39	19	14
sam10	B	T2	5	2	20	41	16	16
sam11	B	T1	3	5	25	29	19	17
sam12	B	T2	2	1	18	16	15	13

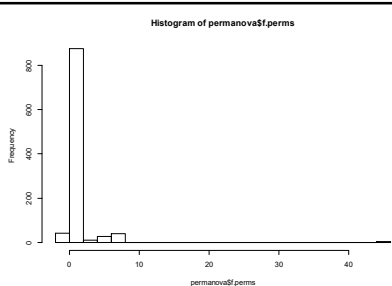
- What if you ignore area?
 - No treatment effect, most variability unexplained

adonis(formula = community ~ treatment)

Permutation: free
Number of permutations: 999

Terms added sequentially (first to last)

	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
treatment	1	0.05133	0.051327	0.64411	0.06051	0.565
Residuals	10	0.79687	0.079687		0.93949	
Total	11	0.84820			1.00000	



- Histogram of permuted pseudo-f
 - hist(permanova\$f.perms)

Strata

	area	treatment	sp1	sp2	sp3	sp4	sp5	sp6
sam1	A	T1	35	32	15	2	10	4
sam2	A	T2	27	20	14	4	13	2
sam3	A	T1	29	34	26	3	20	5
sam4	A	T2	17	26	22	5	11	4
sam5	A	T1	32	29	24	3	16	4
sam6	A	T2	16	19	21	6	9	3
sam7	B	T1	5	4	22	27	20	18
sam8	B	T2	1	2	28	18	14	12
sam9	B	T1	6	4	15	39	19	14
sam10	B	T2	5	2	20	41	16	16
sam11	B	T1	3	5	25	29	19	17
sam12	B	T2	2	1	18	16	15	13

- What if you include area in the model?
 - Marginal treatment effect, most variability explained by area

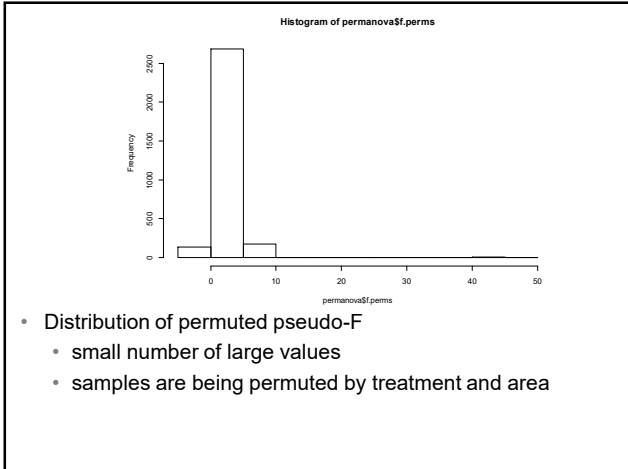
adonis(formula = community ~ treatment * area)

Permutation: free
Number of permutations: 999

Terms added sequentially (first to last)

	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
treatment	1	0.05133	0.05133	4.845	0.06051	0.068
area	1	0.69395	0.69395	65.500	0.81816	0.001 ***
treatment:area	1	0.01816	0.01816	1.714	0.02141	0.158
Residuals	8	0.08476	0.01059		0.09993	
Total	11	0.84820			1.00000	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1



- If our question is whether or not treatments differ while controlling for area, we can control permutations this way.

```
adonis(formula = community ~ treatment, strata = area)
```

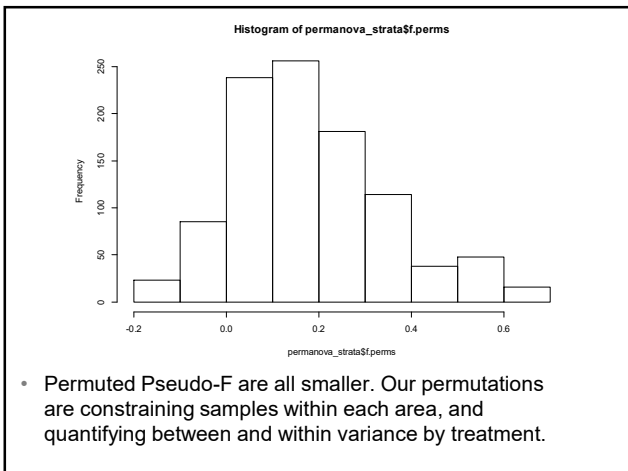
Blocks: strata
Permutation: free
Number of permutations: 999

Terms added sequentially (first to last)

	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
treatment	1	0.05133	0.051327	0.64411	0.06051	0.014 *
Residuals	10	0.79687	0.079687			0.93949
Total	11	0.84820				1.00000

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

- Pseudo-F is smaller, % variance explained by treatment is the same, P is different.



Strata also applies to ANOSIM and MRPP

```
mpp(dat = community, grouping = treatment, distance = "bray")
```

Dissimilarity index: bray
Weights for groups: n

	T1	T2
delta	0.3518	0.3548
n	6	6

Chance corrected within-group agreement A: -0.008071
Based on observed delta 0.3533 and expected delta 0.3505

Significance of delta: 0.489
Permutation: free
Number of permutations: 999

```
mpp(dat = community, grouping = treatment, strata = area)
```

Dissimilarity index: euclidean
Weights for groups: n

	T1	T2
delta	35.12	28.84
n	6	6

Chance corrected within-group agreement A: -0.01142
Based on observed delta 31.98 and expected delta 31.61

Significance of delta: 0.022
Blocks: strata
Permutation: free
Number of permutations: 999

AMOVA

- Very similar to analysis of variance using distance metric (in fact, using a Euclidean distance matrix should yield identical results).
- First developed for analyzing mtDNA haplotypes
 - Distance matrix was pairwise steps in a network
 - Grouping variable was population



Excoffier, L., P.E. Smouse, and J. M. Quattro. 1992. Analysis of molecular variance inferred from metric distances among DNA haplotypes: application to human mitochondrial DNA restriction data. *Genetics*, 131(2):479-491.

AMOVA

- Two amova functions, both named "amova"
- Package ade4
 - Haplotypic approach – requires 1) matrix of haplotype frequency by population, 2) Euclidian distance among haplotypes in network, and 3) assignment of populations to groups
 - Separate function `randtest` tests significance through permutation.
- Package pegas
 - More general approach – requires genetic distance matrix and a factor.

Reading

- Sample script and dataset
- Papers
 - Caspers, B.A., F.C. Schroeder, S. Franke, W. Streich and C.C. Voigt. 2009. Odour-based species recognition in two sympatric species of sac-winged bats (*Saccopteryx bilineata*, *S. laptura*): combining chemical analyses, behavioral observations and odour preference tests. *Behavior Ecology and Sociobiology* **63**: 741-749.
 - Robidoux, M., P. Giorgio, and A. Derry. 2015. Effects of humic stress on the zooplankton from clear and DOC-rich lakes. *Freshwater biology* **60**: 1263-1278.
- Text: Chapter 3, information on similarity measures
- For information on formulas in R, review chapter 11 -`help.start()`

Assignment

- New dataset (`spaeth.csv`) – note that the factors you will use are in the first few columns.
- Rarefy dataset to 50 individuals per sample (`rrarefy`)
- Perform ANOSIM and MRP to test for community differences
 - Test for year and creek differences on proportional data after rarefaction
- Multivariate Analysis of Variance for Distance Matrix
 - Use proportional after rarefaction
 - Test for year and creek differences as above, but also include an interaction term
 - Test for year and season differences while only doing randomizations within creek

Functions to Calculate Distances

- Vegan package
 - `vegdist` – most common community metrics
 - `designdist` – make your own, provide a formula
 - `treedist` – pairwise distance from a dendrogram
- Ape package
 - `dist.dna` – sequence similarity for various evolutionary models
 - `dist.nodes` – phylogenetic distance from a tree
- Labdsv
 - `dsvdis` – community metrics, additional presence/absence
- Base package
 - `dist` – general distances (Euclidian, manhattan etc.) used more broadly.
 - `cor` – correlation matrix
 - `cov` – covariance matrix

Import/Exporting a distance matrix

- Distance matrices are triangular
 - `dist_values<-vegdist(community, method='bray')`
- Writing `dist_values` to a file will generate an error. Use function `as.matrix` to convert to square matrix.
- Note that the diagonal has 0.0, indicating this is dissimilarity

```
as.matrix(dist_values)
      sam1  sam2  sam3  sam4  sam5  sam6  sam7  sam8  sam9
sam1  0.000000  0.1573034  0.14418605  0.19125683  0.10679612  0.25581395  0.58762887  0.6069364  0.57948718
sam2  0.1573034  0.0000000  0.19796954  0.17575758  0.15957447  0.16883117  0.52272727  0.5354839  0.51412429
sam3  0.14418605  0.1979695  0.00000000  0.17821782  0.06666667  0.25654450  0.44600939  0.4687500  0.51401869
sam4  0.19125683  0.1757576  0.17821782  0.00000000  0.13989637  0.08176101  0.43646409  0.4375000  0.50549451
sam5  0.10679612  0.1595745  0.06666667  0.13989637  0.00000000  0.21978022  0.47058824  0.4754098  0.53170732
sam6  0.2558140  0.1688312  0.25654450  0.08176101  0.21978022  0.00000000  0.43529412  0.4362416  0.49707602
sam7  0.5876289  0.5227273  0.44600939  0.43646409  0.47058824  0.43529412  0.00000000  0.1929825  0.12953368
sam8  0.6069364  0.5354839  0.46875000  0.43750000  0.47540984  0.43624161  0.19298246  0.0000000  0.27906977
sam9  0.5794872  0.5141243  0.51401869  0.50549451  0.53170732  0.49707602  0.12953368  0.2790698  0.0000000
```

```
* write.csv(as.matrix(dist_values),"filename.csv")
```

- The saved file is a rectangular matrix, how do you load it and convert back to distance?

```
* rect_matrix<-read.csv("filename.csv", header=T, row.names=1)
* dist_matrix<-as.dist(rect_matrix)
```

```
      sam1  sam2  sam3  sam4  sam5  sam6  sam7  sam8  sam9
sam2  0.15730337
sam3  0.14418605  0.19796954
sam4  0.19125683  0.17575758  0.17821782
sam5  0.10679612  0.15957447  0.06666667  0.13989637
sam6  0.25581395  0.16883117  0.25654450  0.08176101  0.21978022
sam7  0.58762887  0.52272727  0.44600939  0.43646409  0.47058824  0.43529412
sam8  0.60693642  0.53548387  0.46875000  0.43750000  0.47540984  0.43624161  0.19298246
sam9  0.57948718  0.51412429  0.51401869  0.50549451  0.53170732  0.49707602  0.12953368  0.27906977
sam10 0.61616162  0.55555556  0.5295392  0.49189189  0.51923077  0.48275862  0.12244898  0.23428571  0.07614213
sam11 0.60204082  0.53932584  0.44186047  0.45355191  0.46601942  0.45348837  0.05154639  0.16763006  0.13846154
sam12 0.58282209  0.50344828  0.51648352  0.45333333  0.50289017  0.43884892  0.19254658  0.11428571  0.23456790
```

Matrix math

- Adding, multiplying or dividing a single value to a vector or a matrix works as expected (**scalar**)
 - `> a`
 - `[1] -0.72 1.23 0.05 -0.08 0.11`
 - `> a + 5`
 - `[1] 4.28 6.23 5.05 4.92 5.11`
- If the dimensions match, addition/subtraction is done element by element
 - `> b`
 - `[1] 0.59 0.15 -0.36 -1.13 -0.17`
 - `> a`
 - `[1] -0.72 1.23 0.05 -0.08 0.11`
 - `> a + b`
 - `[1] -0.13 1.38 -0.31 -1.21 -0.06`

Matrix math

- Functions `rbind` and `cbind` will combine rows and columns into a new matrix
 - `rbind(community[1,], community[2,], community[5:7,])`
 - `cbind(community[,1], community[,2], community[,5:7])`
- If the dimensions of matrices match, they can be added, multiplied directly:
 - `a<-matrix(c(1,4,-3,2,0,-1), ncol=3)`
 - `b<-matrix(c(6,7,3,-2,-3,5), ncol=3)`
 - `a+b`
 - `a*b`

$$A = \begin{bmatrix} 1 & -3 & 0 \\ 4 & 2 & -1 \end{bmatrix} \text{ and } B = \begin{bmatrix} 6 & 3 & -3 \\ 7 & -2 & 5 \end{bmatrix} \text{ is } A+B = \begin{bmatrix} 7 & 0 & -3 \\ 11 & 0 & 4 \end{bmatrix}$$

Matrix math

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{bmatrix}$$

- **Square matrix** – row and column dimensions equal (**order**)
 - Symmetric ($a_{ij}=a_{ji}$) vs. antisymmetric
 - Diagonals, upper and lower triagles
- **Rectangular matrix** – not square
- **Identity matrix** – 0 with diagonal of 1

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transposition and matrix multiplication

- Transpose a matrix with function `t`
 - `t(community)`
$$A = \begin{bmatrix} 5 & 7 & 0 \\ 1 & 0 & 4 \end{bmatrix}, \quad A^T = \begin{bmatrix} 5 & 1 \\ 7 & 0 \\ 0 & 4 \end{bmatrix}$$
- Provided they **conform**, you multiply vectors and/or matrices through matrix multiplication. To conform, the number of columns in the first must equal the number of rows in the second.
- The resulting matrix has dimensions of the number of rows in the first by the number of columns in the second.

$$\begin{bmatrix} 1 & -3 & 0 \\ 4 & 2 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -5 \\ 5 \end{bmatrix}$$

Matrix math

- **Matrix multiplication (%*%)** – combination of multiplication and addition
 - Matrix with dimensions a_{ij} multiplied by b_{mn} will yield a matrix with c_{in} and $j=m$ to conform
 - Each element is the sum of the products of the row in the first matrix and the column of the second

$$\sum a_{ij} * b_{mn}$$