

# A BRIEF INTRODUCTION TO VEGAN

---

Gavin L. Simpson

CSEE 2015 • May 20th 2015

## BASIC ORDINATION

---

What is **unconstrained**

First we look for major variation, then relate it to environmental variation

vs. constrained ordination, where we only want to see what can be explained by environmental variables of interest

How well do we explain the main patterns in the species data vs how large are the patterns we can explain with the measured data

- Principal Components Analysis — PCA
- Correspondance Analysis — CA
- Nonmetric Multidimensional Scaling — NMDS

Housekeeping

```
setwd("your/working/dir")
```

```
library("vegan")
```

```
data(dune)
```

```
data(dune.env)
```

Data from: Jongman, R.H.G, ter Braak, C.J.F & van Tongeren, O.F.R. (1987). Data Analysis in Community and Landscape Ecology. Pudoc, Wageningen.

```
dim(dune) # number of samples, species
```

```
[1] 20 30
```

```
head(dune[,1:6])
```

	Achimill	Agrostol	Airaprae	Alopgeni	Anthodor	Bellpere
1	1	0	0	0	0	0
2	3	0	0	2	0	3
3	0	4	0	7	0	2
4	0	8	0	2	0	2
5	2	0	0	0	4	2
6	2	0	0	0	3	0

```
head(dune.env, n=3)
```

	A1	Moisture	Management	Use	Manure
1	2.8	1	SF	Haypastu	4
2	3.5	1	BF	Haypastu	2
3	4.3	2	SF	Haypastu	4

```
summary(dune.env)
```

	A1	Moisture	Management	Use	Manure
Min.	: 2.800	1:7	BF:3	Hayfield:7	0:6
1st Qu.:	3.500	2:4	HF:5	Haypastu:8	1:3
Median :	4.200	4:2	NM:6	Pasture :5	2:4
Mean :	4.850	5:7	SF:6		3:4
3rd Qu.:	5.725				4:3
Max.	:11.500				

PCA finds linear combinations of the variables that explain the largest amounts of variance in the data

```
(pca <- rda(dune))
```

```
Call: rda(X = dune)
```

```
              Inertia Rank
Total              84.12
Unconstrained    84.12   19
Inertia is variance
```

Eigenvalues for unconstrained axes:

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
24.795	18.147	7.629	7.153	5.695	4.333	3.199	2.782

(Showed only 8 of all 19 unconstrained eigenvalues)



Vegan has a wrapper function for doing NMDS ordinations using best practices:

- `metaMDS()`

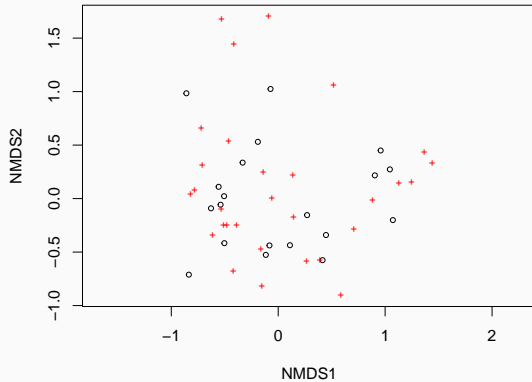
This will do handy things

- standardize your data if necessary
- perform rotation to PCs
- scale coordinates in half change units

```
dune.bray.ord <- metaMDS(dune, distance = "bray", k = 2, trymax = 50)
```

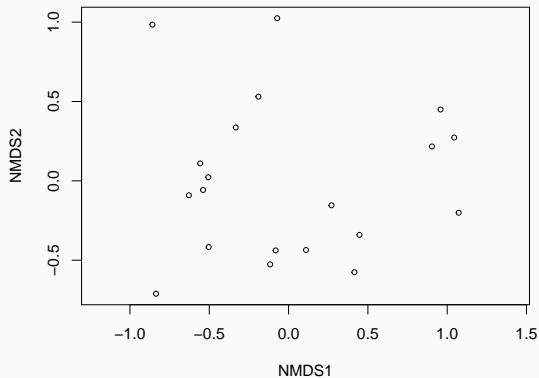
## BASIC ORDINATION AND PLOTTING (USING ALL DEFAULTS)

```
plot(dune.bray.ord)
```



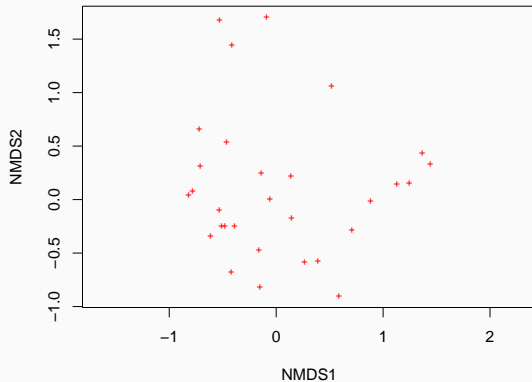
## BASIC ORDINATION AND PLOTTING (JUST PLOTS)

```
plot(dune.bray.ord, display = "sites")
```

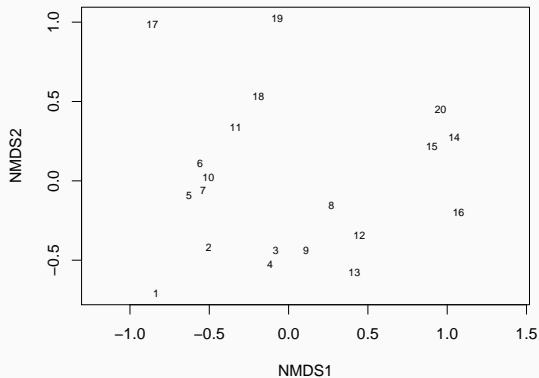


## BASIC ORDINATION AND PLOTTING (JUST SPECIES)

```
plot(dune.bray.ord, display = "species")
```

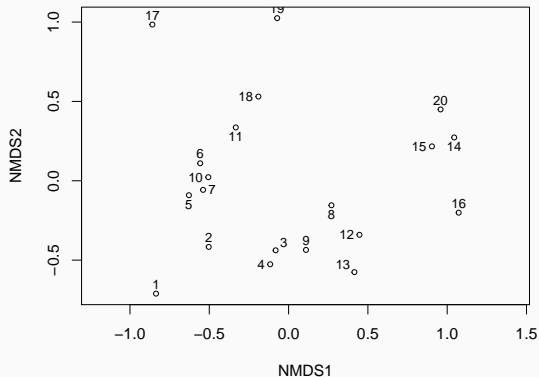


```
plot(dune.bray.ord, display = "sites", type = "text")
```



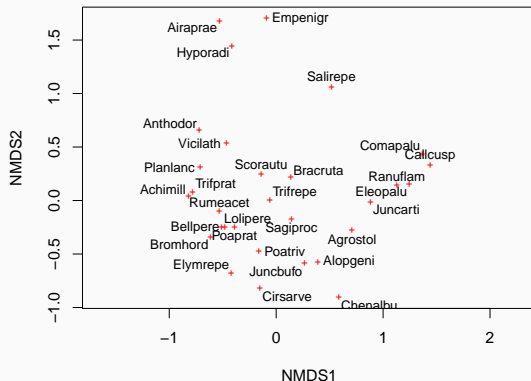
## SITE NAMES INSTEAD OF POINTS

```
plot(dune.bray.ord, display = "sites")  
set.seed(314) ## make reproducible  
ordipointlabel(dune.bray.ord, display = "sites", scaling = 3, add = TRUE)
```



## SITE NAMES INSTEAD OF POINTS

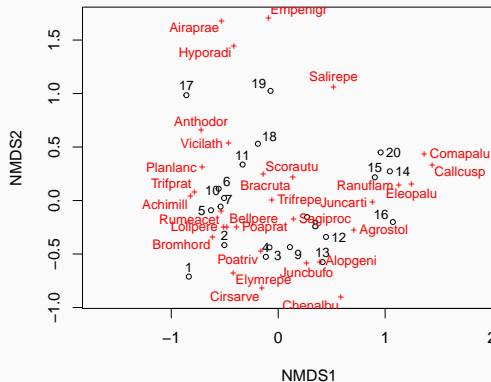
```
plot(dune.bray.ord, display = "species")  
set.seed(314) ## make reproducible  
ordipointlabel(dune.bray.ord, display = "species", scaling = 3, add = TRUE)
```



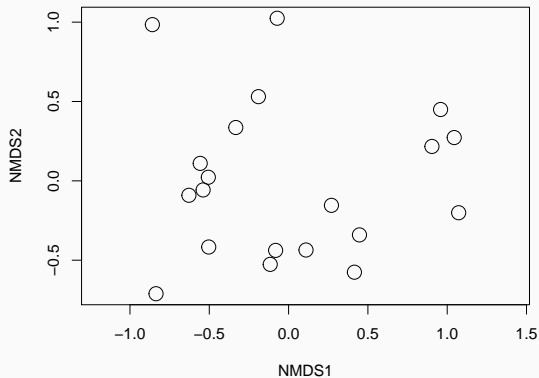


## SITE NAMES INSTEAD OF POINTS

```
plot(dune.bray.ord)
set.seed(314) ## make reproducible
ordipointlabel(dune.bray.ord, scaling = 3, add = TRUE)
```



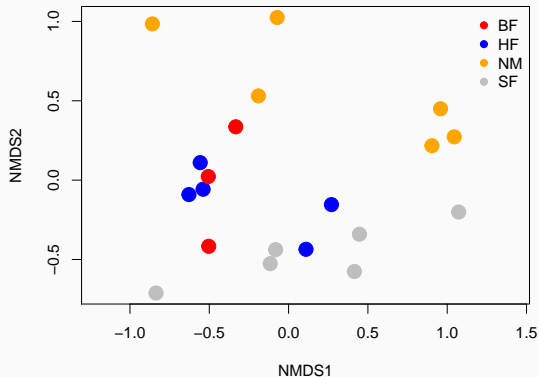
```
plot(dune.bray.ord, display = "sites", cex=2)
```



- Color
- Shape
- Size

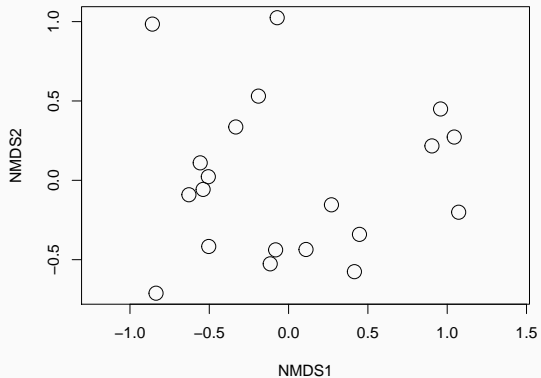
## MODIFYING THE COLOR OF POINTS

```
colors.vec <- c("red", "blue", "orange", "grey")
plot(dune.bray.ord, display = "sites", type = "n")
points(dune.bray.ord, display = "sites", cex=2, pch = 21,
       col = colors.vec[dune.env$Management],
       bg = colors.vec[dune.env$Management])
legend("topright", legend = levels(dune.env$Management), bty = "n",
      col = colors.vec, pch = 21, pt.bg = colors.vec)
```



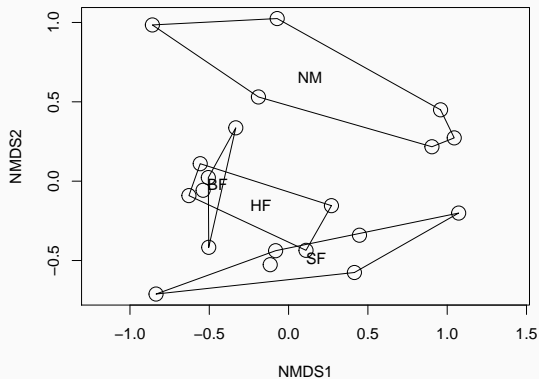
## ADDING OTHER LAYERS

```
plot(dune.bray.ord, display = "sites", cex=2) # just site points
```



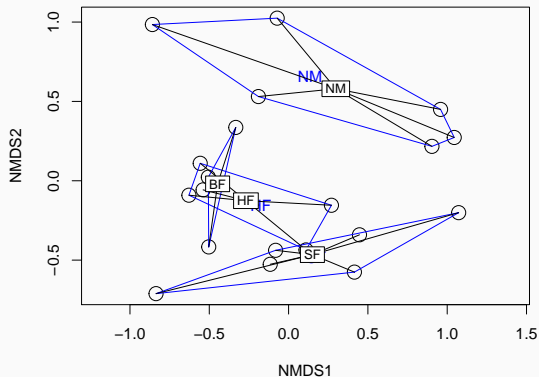
## ADDING OTHER LAYERS

```
plot(dune.bray.ord, display = "sites", cex=2)  
ordihull(dune.bray.ord, groups = dune.env$Management, label = TRUE) # convex hulls
```



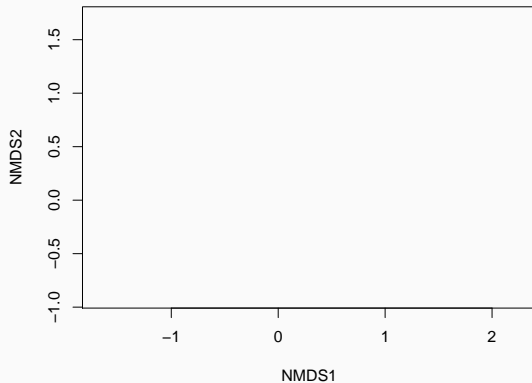
## ADDING OTHER LAYERS

```
plot(dune.bray.ord, display = "sites", cex=2)  
ordihull(dune.bray.ord, groups = dune.env$Management, label = TRUE, col = "blue")  
ordispider(dune.bray.ord, groups = dune.env$Management, label = TRUE)
```



## ADDING OTHER LAYERS - AXES SCALING

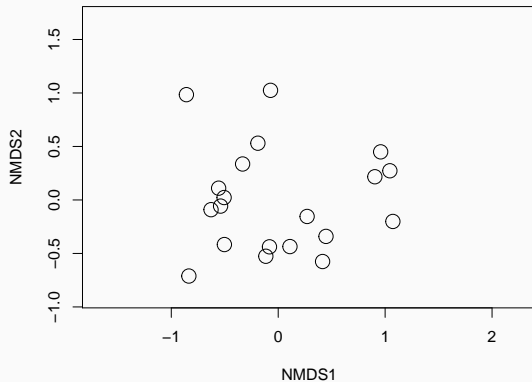
```
plot(dune.bray.ord, type = "n")
```





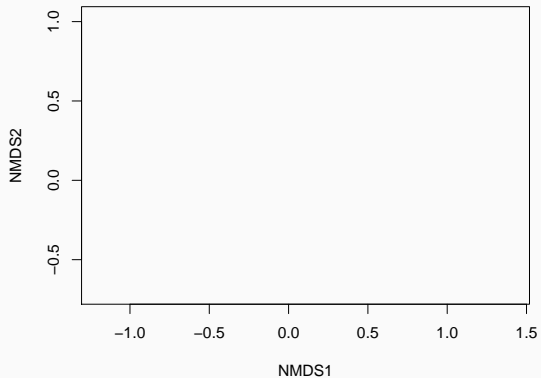
## ADDING OTHER LAYERS - AXES SCALING

```
plot(dune.bray.ord, type = "n")  
points(dune.bray.ord, display = "sites", cex = 2)
```



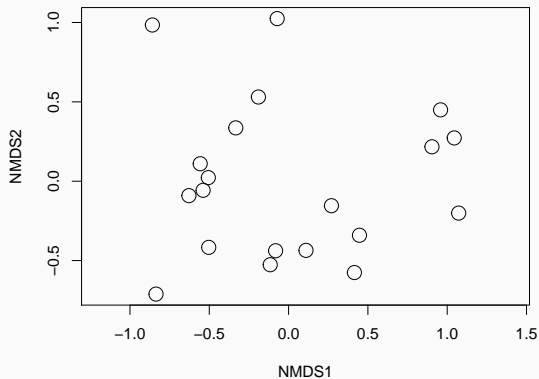
## ADDING OTHER LAYERS - AXES SCALING

```
plot(dune.bray.ord, display = "sites", type = "n")
```



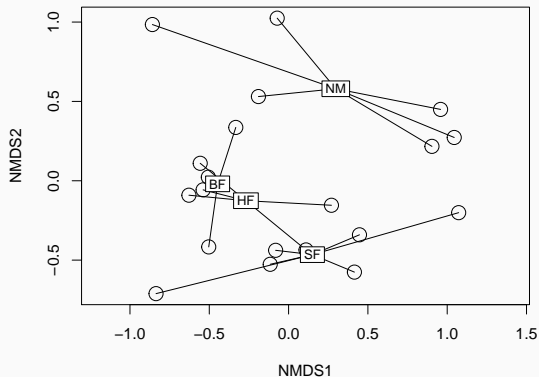
## ADDING OTHER LAYERS - AXES SCALING

```
plot(dune.bray.ord, display = "sites", type = "n")  
points(dune.bray.ord, display = "sites", cex = 2)
```



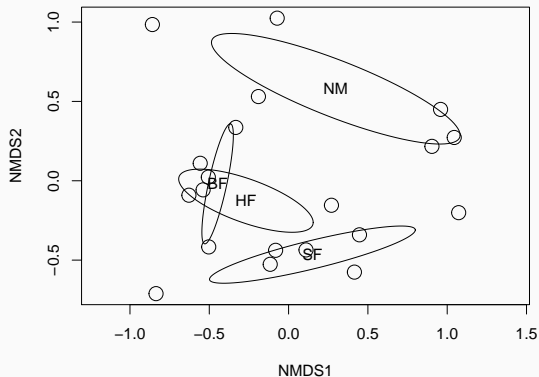
## ADDING OTHER LAYERS

```
plot(dune.bray.ord, display = "sites", type = "n")  
points(dune.bray.ord, display = "sites", cex = 2)  
ordispider(dune.bray.ord, groups = dune.env$Management, label = TRUE)
```



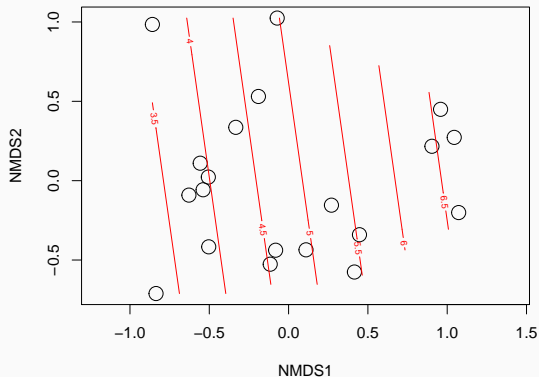
## ADDING OTHER LAYERS

```
plot(dune.bray.ord, display = "sites", type = "n")  
points(dune.bray.ord, display = "sites", cex = 2)  
ordiellipse(dune.bray.ord, groups = dune.env$Management, label = TRUE)
```



## ADDING OTHER LAYERS

```
plot(dune.bray.ord, display = "sites", type = "n")  
points(dune.bray.ord, display = "sites", cex = 2)  
ordisurf(dune.bray.ord, dune.env$A1, add = TRUE)
```



```
dune.bray.ord.A1.fit <- envfit(dune.bray.ord,dune.env$A1, permutations = 1000)
dune.bray.ord.A1.fit
```

\*\*\*VECTORS

```
      NMDS1   NMDS2    r2 Pr(>r)
[1,] 0.99008 0.14052 0.3798 0.01698 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Permutation: free
Number of permutations: 1000
```

## VECTORS IN ORDINATION SPACE

```
plot(dune.bray.ord, display = "sites", type = "n")  
points(dune.bray.ord, display = "sites", cex = 2)  
plot(dune.bray.ord.A1.fit, add = TRUE)  
ordisurf(dune.bray.ord, dune.env$A1, add = TRUE)
```

