Lecture 18 - Class Activity: NMDS and PERMANOVA

Bill Perry

# Lecture 18: Non-metric Multidimensional Scaling (NMDS) and PERMANOVA

## What is NMDS?

NMDS (Non-metric Multidimensional Scaling) is an ordination technique that: - Visualizes dissimilarity between objects in reduced dimensions - Preserves rank order of distances, not exact distances - Works well with non-linear ecological relationships - Makes few assumptions about data structure

## When to Use NMDS

Use NMDS when you have: - **Community data**: Species abundance or presence/absence matrices - **Non-linear relationships**: When PCA assumptions are violated - **Complex ecological gradients**: Multiple environmental factors affecting communities

## Key Concepts of NMDS

1. **Dissimilarity matrices** instead of covariance
2. **Stress values** measure goodness of fit (<0.2 is acceptable)
3. **Iterative algorithm** to find optimal configuration
4. **No eigenvalues** - axes have no inherent meaning
5. **Rank-based** - preserves order, not exact distances

|  |  |
| --- | --- |
|  | **Critical First Step**  Always check your stress value! Stress < 0.1 is excellent, 0.1-0.2 is good, > 0.2 is poor representation. |

# Part 1: Data Preparation and Exploration

## Load and Prepare Data

We’ll use the iris dataset for this analysis, treating it as if the measurements represent abundances of different “species” at different “sites”.

# Load iris data  
iris\_df <- read\_csv("data/iris.csv") %>%  
 clean\_names()

Rows: 150 Columns: 5  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (1): species  
dbl (4): sepal\_length, sepal\_width, petal\_length, petal\_width  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# View data structure  
head(iris\_df)

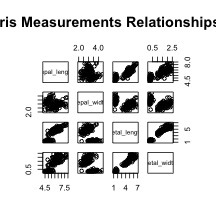
# A tibble: 6 × 5  
 sepal\_length sepal\_width petal\_length petal\_width species  
 <dbl> <dbl> <dbl> <dbl> <chr>   
1 5.1 3.5 1.4 0.2 setosa   
2 4.9 3 1.4 0.2 setosa   
3 4.7 3.2 1.3 0.2 setosa   
4 4.6 3.1 1.5 0.2 setosa   
5 5 3.6 1.4 0.2 setosa   
6 5.4 3.9 1.7 0.4 setosa

# For NMDS, we need just the numeric columns (our "species" data)  
# We'll keep species as our grouping variable  
iris\_species\_df <- iris\_df %>%  
 dplyr::select(species)  
  
iris\_numeric\_df <- iris\_df %>%  
 dplyr::select(-species)  
  
# Check the structure  
str(iris\_numeric\_df)

tibble [150 × 4] (S3: tbl\_df/tbl/data.frame)  
 $ sepal\_length: num [1:150] 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  
 $ sepal\_width : num [1:150] 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  
 $ petal\_length: num [1:150] 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  
 $ petal\_width : num [1:150] 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...

## Visualize the Data

# Create a pairs plot to see relationships  
iris\_pairs\_plot <- iris\_df %>%  
 dplyr::select(-species) %>%  
 pairs(main = "Iris Measurements Relationships")



# Part 2: Running NMDS

## Step 1: Calculate Distance Matrix

# Calculate Bray-Curtis distance (common for ecological data)  
# For iris data, we'll use Euclidean distance since these are measurements  
iris\_dist <- dist(iris\_numeric\_df, method = "euclidean")  
  
# Check the first few distances  
iris\_dist[1:5]

[1] 0.5385165 0.5099020 0.6480741 0.1414214 0.6164414

## Step 2: Run NMDS

# Run NMDS with 2 dimensions  
set.seed(123) # For reproducibility  
iris\_nmds\_model <- metaMDS(iris\_numeric\_df,   
 distance = "euclidean",  
 k = 2, # Number of dimensions  
 trymax = 100) # Maximum iterations

Run 0 stress 0.02525035   
Run 1 stress 0.04045544   
Run 2 stress 0.03566855   
Run 3 stress 0.0268287   
Run 4 stress 0.03695952   
Run 5 stress 0.03104439   
Run 6 stress 0.02682881   
Run 7 stress 0.03962709   
Run 8 stress 0.03210416   
Run 9 stress 0.04325555   
Run 10 stress 0.02525031   
... New best solution  
... Procrustes: rmse 1.741237e-05 max resid 7.129696e-05   
... Similar to previous best  
Run 11 stress 0.02525038   
... Procrustes: rmse 1.217807e-05 max resid 8.876481e-05   
... Similar to previous best  
Run 12 stress 0.04000578   
Run 13 stress 0.03552087   
Run 14 stress 0.0309981   
Run 15 stress 0.04078306   
Run 16 stress 0.04008785   
Run 17 stress 0.03197035   
Run 18 stress 0.02525033   
... Procrustes: rmse 5.481604e-06 max resid 2.712782e-05   
... Similar to previous best  
Run 19 stress 0.02682875   
Run 20 stress 0.03270101   
\*\*\* Best solution repeated 3 times

# Check the results  
iris\_nmds\_model

Call:  
metaMDS(comm = iris\_numeric\_df, distance = "euclidean", k = 2, trymax = 100)   
  
global Multidimensional Scaling using monoMDS  
  
Data: iris\_numeric\_df   
Distance: euclidean   
  
Dimensions: 2   
Stress: 0.02525031   
Stress type 1, weak ties  
Best solution was repeated 3 times in 20 tries  
The best solution was from try 10 (random start)  
Scaling: centring, PC rotation   
Species: expanded scores based on 'iris\_numeric\_df'

**Interpretation**: - Stress value: 0.025 - This is excellent representation

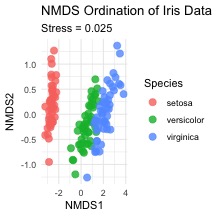
## Step 3: Extract NMDS Scores

# Extract NMDS scores for plotting  
nmds\_scores\_df <- as.data.frame(iris\_nmds\_model$points) %>%  
 rename(nmds1 = MDS1, nmds2 = MDS2) %>%  
 bind\_cols(iris\_species\_df)  
  
# View the scores  
head(nmds\_scores\_df)

nmds1 nmds2 species  
1 -2.687311 0.2758924 setosa  
2 -2.717262 -0.1705943 setosa  
3 -2.882071 -0.1017541 setosa  
4 -2.746720 -0.2603709 setosa  
5 -2.730486 0.2902002 setosa  
6 -2.311408 0.6625723 setosa

## Step 4: Create NMDS Plot

# Basic NMDS plot  
nmds\_basic\_plot <- ggplot(nmds\_scores\_df, aes(x = nmds1, y = nmds2, color = species)) +  
 geom\_point(size = 3, alpha = 0.8) +  
 labs(title = "NMDS Ordination of Iris Data",  
 subtitle = paste("Stress =", round(iris\_nmds\_model$stress, 3)),  
 x = "NMDS1",   
 y = "NMDS2",  
 color = "Species") +  
 theme\_minimal()  
  
nmds\_basic\_plot

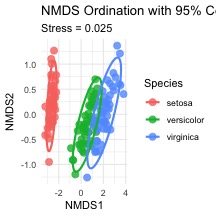


## Step 5: Add Confidence Ellipses

# NMDS plot with ellipses  
nmds\_ellipse\_plot <- ggplot(nmds\_scores\_df, aes(x = nmds1, y = nmds2, color = species)) +  
 geom\_point(size = 3, alpha = 0.8) +  
 stat\_ellipse(level = 0.95, size = 1) +  
 labs(title = "NMDS Ordination with 95% Confidence Ellipses",  
 subtitle = paste("Stress =", round(iris\_nmds\_model$stress, 3)),  
 x = "NMDS1",   
 y = "NMDS2",  
 color = "Species") +  
 theme\_minimal()

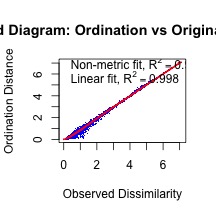
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
ℹ Please use `linewidth` instead.

nmds\_ellipse\_plot



## Step 6: Stress Plot (Shepard Diagram)

# Create stress plot to evaluate fit  
stressplot(iris\_nmds\_model, main = "Shepard Diagram: Ordination vs Original Distances")



# Part 3: PERMANOVA Analysis

## What is PERMANOVA?

PERMANOVA (Permutational Multivariate Analysis of Variance) tests whether groups have different multivariate centroids using permutation tests.

## Step 1: Run PERMANOVA

# Run PERMANOVA to test if species differ in multivariate space  
set.seed(456)  
iris\_permanova\_model <- adonis2(iris\_numeric\_df ~ species,   
 data = iris\_df,  
 method = "euclidean",  
 permutations = 999)  
  
# View results  
iris\_permanova\_model

Permutation test for adonis under reduced model  
Permutation: free  
Number of permutations: 999  
  
adonis2(formula = iris\_numeric\_df ~ species, data = iris\_df, permutations = 999, method = "euclidean")  
 Df SumOfSqs R2 F Pr(>F)   
Model 2 592.07 0.86894 487.33 0.001 \*\*\*  
Residual 147 89.30 0.13106   
Total 149 681.37 1.00000   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**Interpretation**: - F-statistic: 487.33 - R² (variance explained): 0.869 - p-value: 0.001

## Step 2: Check Homogeneity of Dispersions

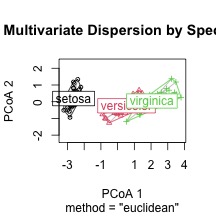
Before interpreting PERMANOVA, we need to check if groups have similar multivariate spread.

# Test homogeneity of multivariate dispersions  
iris\_dist\_full <- dist(iris\_numeric\_df)  
dispersion\_model <- betadisper(iris\_dist\_full, iris\_df$species)  
  
# Test for differences in dispersion  
dispersion\_test <- anova(dispersion\_model)  
dispersion\_test

Analysis of Variance Table  
  
Response: Distances  
 Df Sum Sq Mean Sq F value Pr(>F)   
Groups 2 2.9092 1.45458 10.748 4.4e-05 \*\*\*  
Residuals 147 19.8941 0.13533   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Step 3: Visualize Dispersions

# Plot dispersions  
plot(dispersion\_model, main = "Multivariate Dispersion by Species")



## Step 4: Pairwise PERMANOVA

# Function for pairwise PERMANOVA comparisons  
pairwise\_permanova <- function(data\_matrix, groups, distance\_method = "euclidean") {  
 # Get unique group combinations  
 group\_levels <- unique(groups)  
 comparisons <- combn(group\_levels, 2)  
   
 # Initialize results  
 results\_df <- data.frame(  
 group1 = character(),  
 group2 = character(),  
 f\_statistic = numeric(),  
 r\_squared = numeric(),  
 p\_value = numeric()  
 )  
   
 # Run pairwise comparisons  
 for(i in 1:ncol(comparisons)) {  
 # Subset data  
 group1 <- comparisons[1, i]  
 group2 <- comparisons[2, i]  
 subset\_indices <- which(groups %in% c(group1, group2))  
   
 subset\_data <- data\_matrix[subset\_indices, ]  
 subset\_groups <- groups[subset\_indices]  
   
 # Run PERMANOVA  
 temp\_result <- adonis2(subset\_data ~ subset\_groups,   
 method = distance\_method,  
 permutations = 999)  
   
 # Store results  
 results\_df <- rbind(results\_df, data.frame(  
 group1 = group1,  
 group2 = group2,  
 f\_statistic = temp\_result$F[1],  
 r\_squared = temp\_result$R2[1],  
 p\_value = temp\_result$"Pr(>F)"[1]  
 ))  
 }  
   
 # Adjust p-values for multiple comparisons  
 results\_df$p\_adjusted <- p.adjust(results\_df$p\_value, method = "bonferroni")  
   
 return(results\_df)  
}  
  
# Run pairwise comparisons  
pairwise\_results\_df <- pairwise\_permanova(iris\_numeric\_df, iris\_df$species)  
pairwise\_results\_df

group1 group2 f\_statistic r\_squared p\_value p\_adjusted  
1 setosa versicolor 551.0039 0.8489994 0.001 0.003  
2 setosa virginica 943.7992 0.9059320 0.001 0.003  
3 versicolor virginica 86.7697 0.4696100 0.001 0.003

# Part 4: ANOSIM Analysis

## What is ANOSIM?

ANOSIM (Analysis of Similarities) tests whether there is a significant difference between groups using rank dissimilarities.

## Step 1: Run ANOSIM

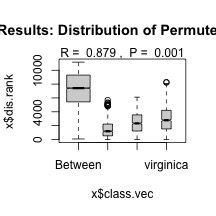
# Run ANOSIM  
set.seed(789)  
iris\_anosim\_model <- anosim(iris\_dist\_full, iris\_df$species, permutations = 999)  
  
# View results  
iris\_anosim\_model

Call:  
anosim(x = iris\_dist\_full, grouping = iris\_df$species, permutations = 999)   
Dissimilarity: euclidean   
  
ANOSIM statistic R: 0.8794   
 Significance: 0.001   
  
Permutation: free  
Number of permutations: 999

**Interpretation**: - R statistic: 0.879 - p-value: 0.001 - R close to 1 indicates strong separation between groups

## Step 2: Plot ANOSIM Results

# Plot ANOSIM results  
plot(iris\_anosim\_model, main = "ANOSIM Results: Distribution of Permuted R Statistics")



# Part 5: Environmental Fitting (Optional)

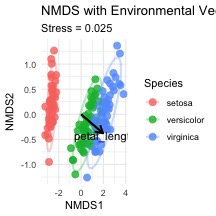
If we had environmental variables, we could fit them to the ordination.

# For demonstration, let's use petal\_length as an "environmental" variable  
env\_data\_df <- data.frame(petal\_length = iris\_df$petal\_length)  
  
# Fit environmental vector  
env\_fit\_model <- envfit(iris\_nmds\_model, env\_data\_df, permutations = 999)  
env\_fit\_model

\*\*\*VECTORS  
  
 NMDS1 NMDS2 r2 Pr(>r)   
petal\_length 0.98205 -0.18862 0.9979 0.001 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
Permutation: free  
Number of permutations: 999

## Visualize Environmental Vectors

# Extract vector coordinates  
env\_coords\_df <- as.data.frame(env\_fit\_model$vectors$arrows \* 2) # Scale for visibility  
env\_coords\_df$variable <- rownames(env\_coords\_df)  
  
# NMDS plot with environmental vector  
nmds\_env\_plot <- ggplot(nmds\_scores\_df, aes(x = nmds1, y = nmds2, color = species)) +  
 geom\_point(size = 3, alpha = 0.8) +  
 stat\_ellipse(level = 0.95, size = 1, alpha = 0.3) +  
 geom\_segment(data = env\_coords\_df,  
 aes(x = 0, y = 0, xend = NMDS1, yend = NMDS2),  
 arrow = arrow(length = unit(0.3, "cm")),  
 color = "black", size = 1) +  
 geom\_text(data = env\_coords\_df,  
 aes(x = NMDS1 \* 1.1, y = NMDS2 \* 1.1, label = variable),  
 color = "black", size = 4) +  
 labs(title = "NMDS with Environmental Vector",  
 subtitle = paste("Stress =", round(iris\_nmds\_model$stress, 3)),  
 x = "NMDS1",   
 y = "NMDS2",  
 color = "Species") +  
 theme\_minimal()  
  
nmds\_env\_plot



# Summary Checklist for NMDS and PERMANOVA

|  |  |
| --- | --- |
|  | **Analysis Checklist**   1. **Prepare your data** - ensure numeric matrix format 2. **Choose appropriate distance measure**    * Bray-Curtis for abundance data    * Euclidean for measurement data 3. **Run NMDS** with sufficient iterations 4. **Check stress value** - must be < 0.2 5. **Create ordination plots** with groups identified 6. **Test homogeneity of dispersions** before PERMANOVA 7. **Run PERMANOVA** to test group differences 8. **Consider ANOSIM** as complementary test 9. **Fit environmental variables** if available |

## Key Points to Remember

* **NMDS preserves rank order** of distances, not exact values
* **Stress < 0.2** is acceptable, < 0.1 is excellent
* **PERMANOVA tests centroids**, ANOSIM tests overlap
* **Check dispersion homogeneity** - violated assumption affects interpretation
* **Multiple comparisons** require p-value adjustment
* **Axes have no inherent meaning** in NMDS (unlike PCA)
* **Use appropriate distance measures** for your data type

|  |  |
| --- | --- |
|  | **Key Takeaways from NMDS/PERMANOVA Analysis**   1. **NMDS is flexible** - works with any distance measure and makes few assumptions 2. **Stress indicates fit quality** - always report and check this value 3. **PERMANOVA is powerful** but assumes homogeneous dispersions 4. **ANOSIM is complementary** - provides different perspective on group separation 5. **Visualization is crucial** - always plot your ordination results 6. **Environmental fitting** helps interpret ecological patterns 7. **Permutation tests** avoid distributional assumptions   Remember: NMDS is iterative and may find different solutions - always set a seed for reproducibility! |