Lecture 18 - Multivariate Community Analysis

Bill Perry

# **Lecture 18:** Multivariate Community Analysis

Let’s import the data and start to explore it

# Set theme  
theme\_set(theme\_minimal() +   
 theme(text = element\_text(size = 12),  
 plot.title = element\_text(size = 14, face = "bold")))  
  
# Load data  
doubs\_env <- read\_csv("data/DoubsEnv.csv") %>%  
 rename(site = 1) %>%  
 mutate(site = as.factor(site))  
  
doubs\_spe <- read\_csv("data/DoubsSpe.csv") %>%  
 rename(site = 1) %>%  
 mutate(site = as.factor(site))  
  
# Create river reach groups based on distance from source  
doubs\_env <- doubs\_env %>%  
 mutate(reach = case\_when(  
 das <= 30 ~ "Upper",  
 das <= 80 ~ "Middle",   
 TRUE ~ "Lower"  
 ) %>% factor(levels = c("Upper", "Middle", "Lower")))  
  
doubs\_spe <- doubs\_spe %>%  
 left\_join(doubs\_env %>% select(site, reach), by = "site")  
  
cat("Species data structure:\n")

Species data structure:

str(doubs\_spe)

tibble [29 × 29] (S3: tbl\_df/tbl/data.frame)  
 $ site : Factor w/ 29 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...  
 $ CHA : num [1:29] 0 0 0 0 0 0 0 0 0 1 ...  
 $ TRU : num [1:29] 3 5 5 4 2 3 5 0 1 3 ...  
 $ VAI : num [1:29] 0 4 5 5 3 4 4 1 4 4 ...  
 $ LOC : num [1:29] 0 3 5 5 2 5 5 3 4 1 ...  
 $ OMB : num [1:29] 0 0 0 0 0 0 0 0 0 1 ...  
 $ BLA : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ HOT : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ TOX : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ VAN : num [1:29] 0 0 0 0 5 1 1 0 2 0 ...  
 $ CHE : num [1:29] 0 0 0 1 2 2 1 5 2 1 ...  
 $ BAR : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ SPI : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ GOU : num [1:29] 0 0 0 1 2 1 0 0 1 0 ...  
 $ BRO : num [1:29] 0 0 1 2 4 1 0 0 0 0 ...  
 $ PER : num [1:29] 0 0 0 2 4 1 0 0 0 0 ...  
 $ BOU : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ PSO : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ ROT : num [1:29] 0 0 0 0 2 0 0 0 0 0 ...  
 $ CAR : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ TAN : num [1:29] 0 0 0 1 3 2 0 1 0 0 ...  
 $ BCO : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ PCH : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ GRE : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ GAR : num [1:29] 0 0 0 0 5 1 0 4 0 0 ...  
 $ BBO : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ ABL : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ ANG : num [1:29] 0 0 0 0 0 0 0 0 0 0 ...  
 $ reach: Factor w/ 3 levels "Upper","Middle",..: 1 1 1 1 1 2 2 2 3 3 ...

cat("\nEnvironmental data structure:\n")

Environmental data structure:

str(doubs\_env)

tibble [29 × 13] (S3: tbl\_df/tbl/data.frame)  
 $ site : Factor w/ 29 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...  
 $ das : num [1:29] 0.3 2.2 10.2 18.5 21.5 ...  
 $ alt : num [1:29] 934 932 914 854 849 846 841 752 617 483 ...  
 $ pen : num [1:29] 48 3 3.7 3.2 2.3 3.2 6.6 1.2 9.9 4.1 ...  
 $ deb : num [1:29] 0.84 1 1.8 2.53 2.64 2.86 4 4.8 10 19.9 ...  
 $ pH : num [1:29] 7.9 8 8.3 8 8.1 7.9 8.1 8 7.7 8.1 ...  
 $ dur : num [1:29] 45 40 52 72 84 60 88 90 82 96 ...  
 $ pho : num [1:29] 0.01 0.02 0.05 0.1 0.38 0.2 0.07 0.3 0.06 0.3 ...  
 $ nit : num [1:29] 0.2 0.2 0.22 0.21 0.52 0.15 0.15 0.82 0.75 1.6 ...  
 $ amm : num [1:29] 0 0.1 0.05 0 0.2 0 0 0.12 0.01 0 ...  
 $ oxy : num [1:29] 12.2 10.3 10.5 11 8 10.2 11.1 7.2 10 11.5 ...  
 $ dbo : num [1:29] 2.7 1.9 3.5 1.3 6.2 5.3 2.2 5.2 4.3 2.7 ...  
 $ reach: Factor w/ 3 levels "Upper","Middle",..: 1 1 1 1 1 2 2 2 3 3 ...

# Today’s Data: The Doubs River

## About the Doubs River Dataset

**Study System:**

* Doubs River, France
* 29 sites from upstream to downstream
* Collected by Verneaux (1973)
* Classic community ecology dataset

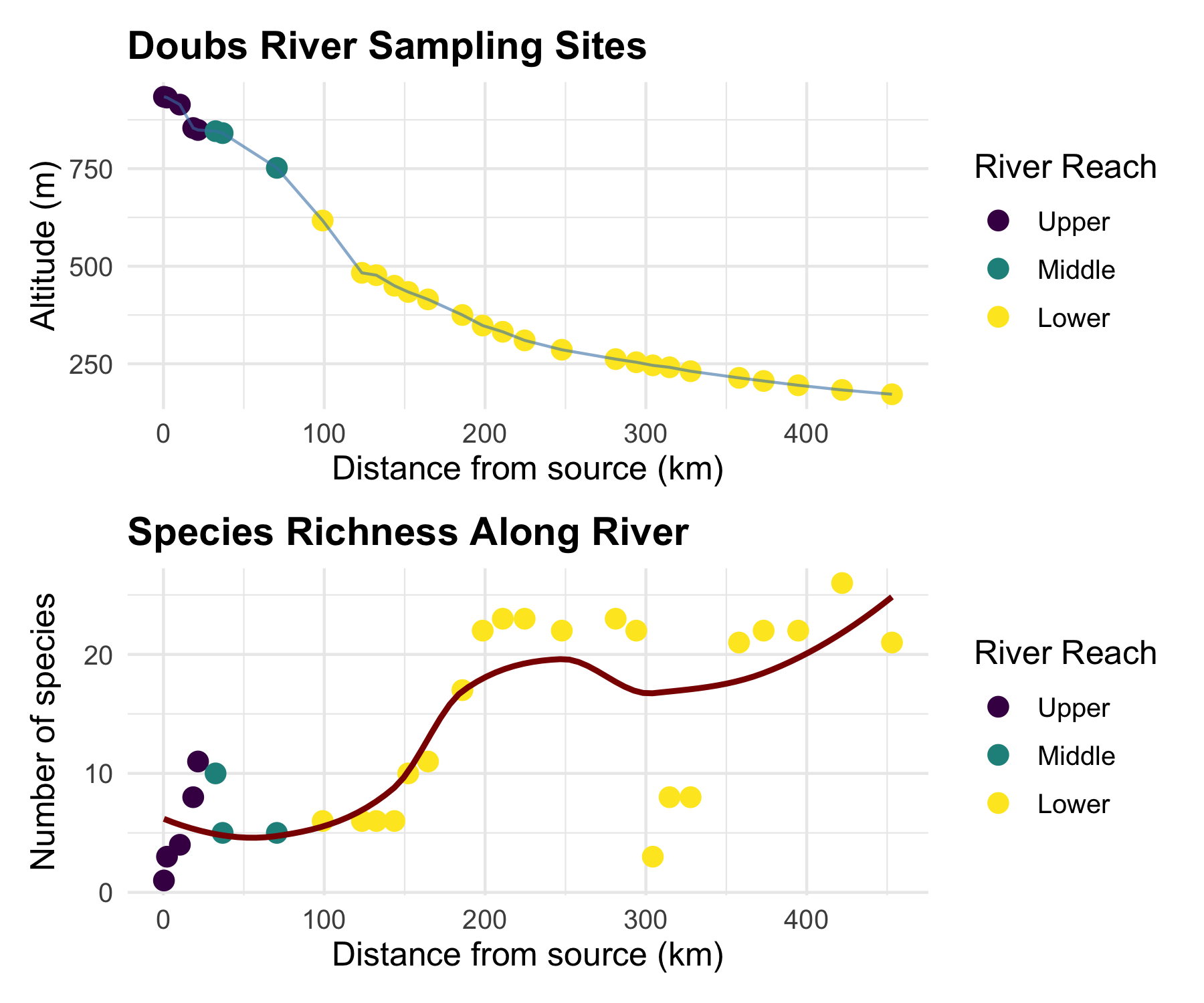
**Two Datasets:**

1. **Environmental**: 11 water quality variables
2. **Species**: 27 fish species abundances (0-5 scale)

**Research Questions:**

* How do fish communities change along the river?
* Which environmental factors drive community composition?
* Are there distinct community types?

# Show the sampling design  
p1 <- doubs\_env %>%  
 ggplot(aes(das, alt, color = reach)) +  
 geom\_point(size = 3) +  
 geom\_line(color = "steelblue", alpha = 0.6) +  
 labs(title = "Doubs River Sampling Sites",  
 x = "Distance from source (km)",  
 y = "Altitude (m)",  
 color = "River Reach") +  
 scale\_color\_viridis\_d()  
  
# Show species richness along river  
species\_richness <- doubs\_spe %>%  
 select(-site, -reach) %>%  
 mutate(richness = rowSums(. > 0)) %>%  
 bind\_cols(doubs\_env %>% select(das, reach))  
  
p2 <- species\_richness %>%  
 ggplot(aes(das, richness, color = reach)) +  
 geom\_point(size = 3) +  
 geom\_smooth(method = "loess", se = FALSE, color = "darkred") +  
 labs(title = "Species Richness Along River",  
 x = "Distance from source (km)",  
 y = "Number of species",  
 color = "River Reach") +  
 scale\_color\_viridis\_d()  
  
p1 / p2



# Non-metric Multidimensional Scaling (NMDS)

## What is NMDS?

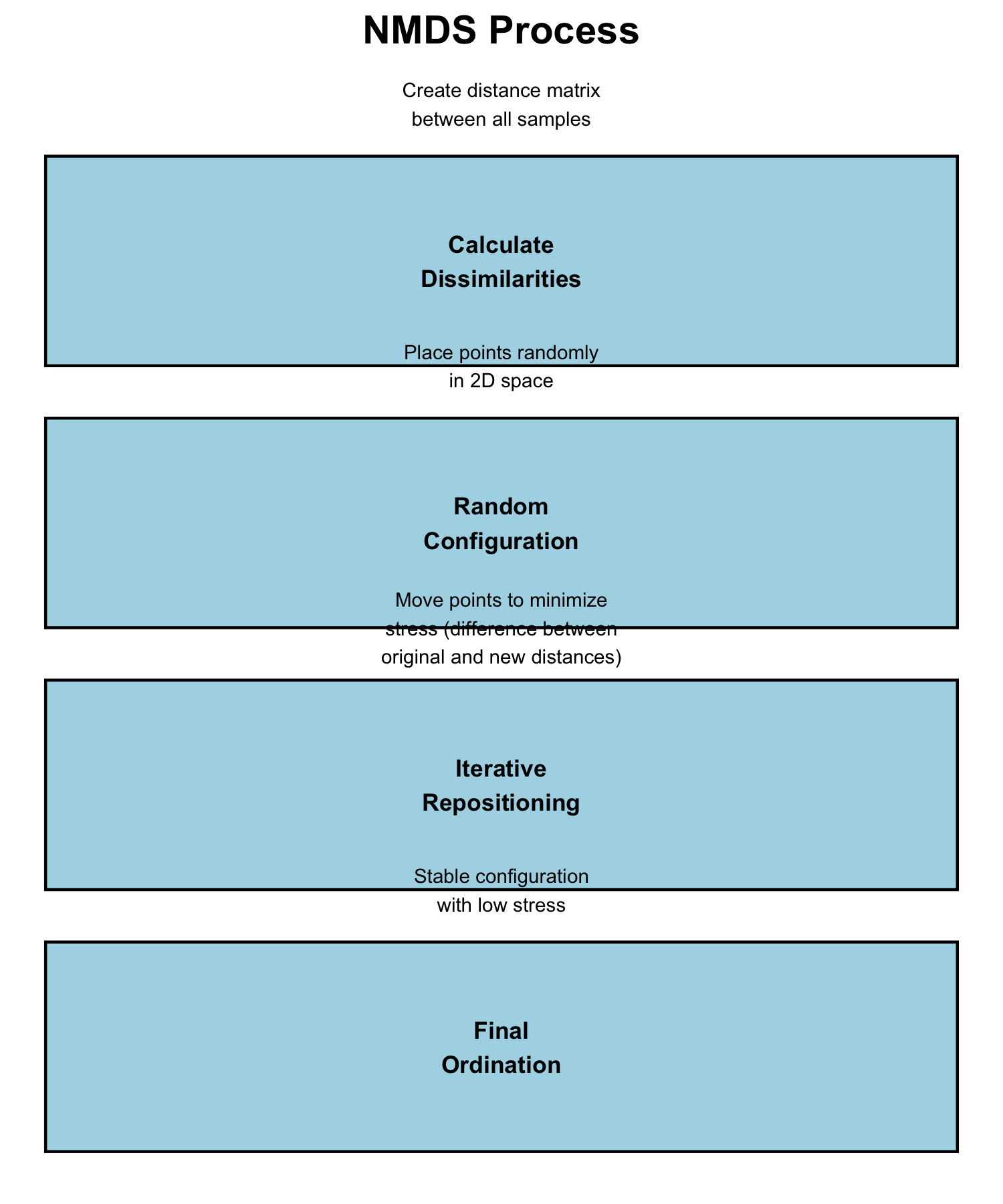
**Overview of NMDS:**

* **Purpose**: Visualize dissimilarity between objects in 2D/3D space
* **Goal**: Preserve rank order of distances, not exact distances
* **Method**: Iterative repositioning to minimize stress
* **Output**: Ordination plot showing relationships between samples

**Key Differences from PCA:**

* Uses dissimilarity matrices (not covariance)
* Non-parametric (rank-based)
* Better for non-linear ecological relationships
* No assumption about data distribution

# Conceptual diagram of NMDS process  
tibble(  
 step = 1:4,  
 process = c("Calculate\nDissimilarities",   
 "Random\nConfiguration",   
 "Iterative\nRepositioning",   
 "Final\nOrdination"),  
 description = c("Create distance matrix\nbetween all samples",  
 "Place points randomly\nin 2D space",  
 "Move points to minimize\nstress (difference between\noriginal and new distances)",  
 "Stable configuration\nwith low stress")  
) %>%  
 ggplot(aes(1, step)) +  
 geom\_rect(aes(xmin = 0.5, xmax = 1.5, ymin = step - 0.4, ymax = step + 0.4),  
 fill = "lightblue", color = "black") +  
 geom\_text(aes(label = process), fontface = "bold", size = 3) +  
 geom\_text(aes(label = description, y = step - 0.6), size = 2.5) +  
 scale\_y\_reverse() +  
 labs(title = "NMDS Process") +  
 theme\_void() +  
 theme(plot.title = element\_text(hjust = 0.5, size = 14, face = "bold"))



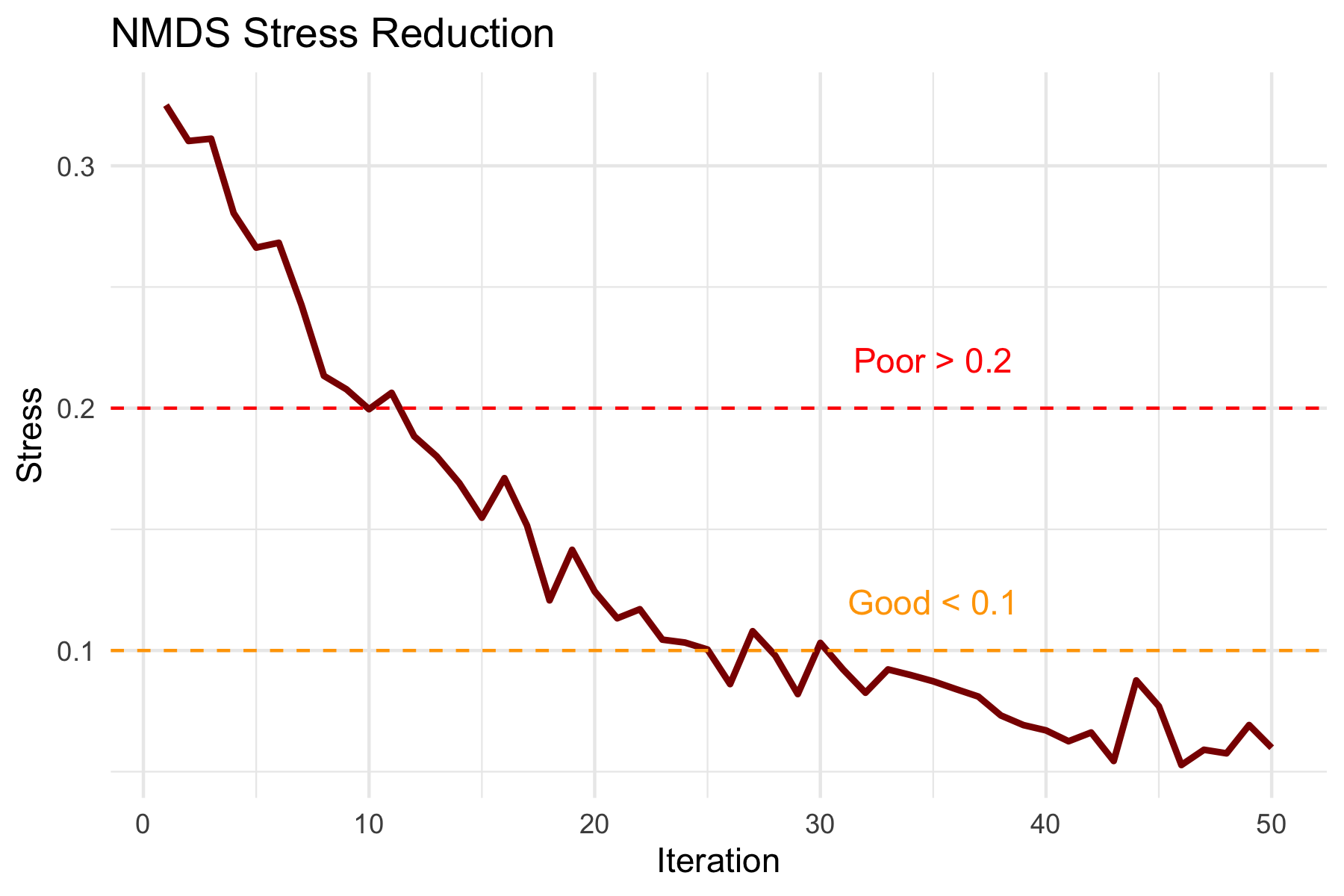
## How NMDS Works

**The NMDS Algorithm:**

1. **Calculate dissimilarity matrix** between all pairs of sites
2. **Start with random configuration** of points in 2D space
3. **Calculate stress** = difference between original distances and ordination distances
4. **Move points** to reduce stress
5. **Repeat** until stress cannot be reduced further
6. **Try multiple random starts** to avoid local minima

**Stress Values:** - < 0.1: Good representation - 0.1-0.2: Acceptable - > 0.2: Poor representation

# Simulate stress reduction over iterations  
set.seed(123)  
iterations <- 1:50  
stress <- 0.3 \* exp(-iterations/15) + 0.05 + rnorm(50, 0, 0.01)  
stress[stress < 0.05] <- 0.05  
  
tibble(iterations, stress) %>%  
 ggplot(aes(iterations, stress)) +  
 geom\_line(color = "darkred", size = 1) +  
 geom\_hline(yintercept = 0.1, linetype = "dashed", color = "orange") +  
 geom\_hline(yintercept = 0.2, linetype = "dashed", color = "red") +  
 annotate("text", x = 35, y = 0.12, label = "Good < 0.1", color = "orange") +  
 annotate("text", x = 35, y = 0.22, label = "Poor > 0.2", color = "red") +  
 labs(title = "NMDS Stress Reduction",  
 x = "Iteration",  
 y = "Stress") +  
 theme\_minimal()



## NMDS Assumptions

**NMDS Assumptions:**

✅ **Few assumptions:**

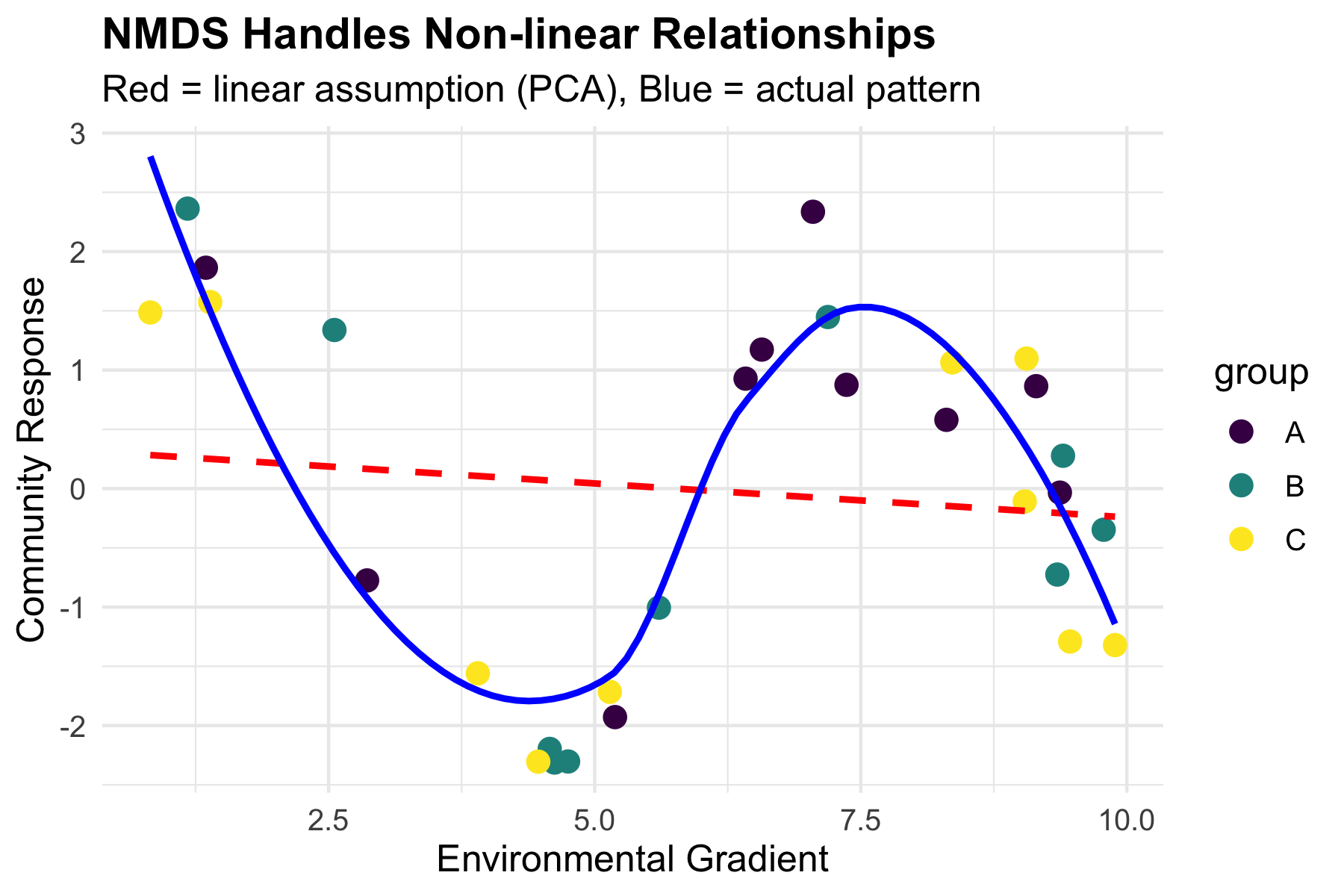
* Samples are independent
* Dissimilarity measure is appropriate
* Sufficient data for stable solution

❌ **No assumptions about:**

* Data distribution
* Linear relationships
* Homoscedasticity
* Normality

**This makes NMDS very robust for ecological data!**

# Show why NMDS is robust - non-linear example  
set.seed(42)  
n <- 30  
x <- runif(n, 0, 10)  
y <- 2 \* sin(x) + rnorm(n, 0, 0.5)  
group <- rep(c("A", "B", "C"), each = 10)  
  
tibble(x, y, group) %>%  
 ggplot(aes(x, y, color = group)) +  
 geom\_point(size = 3) +  
 geom\_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") +  
 geom\_smooth(method = "loess", se = FALSE, color = "blue") +  
 labs(title = "NMDS Handles Non-linear Relationships",  
 subtitle = "Red = linear assumption (PCA), Blue = actual pattern",  
 x = "Environmental Gradient",  
 y = "Community Response") +  
 scale\_color\_viridis\_d()



# NMDS in Practice

## Running NMDS on Fish Communities

# Prepare species data (remove site column)  
spe\_matrix <- doubs\_spe %>%  
 select(-site, -reach) %>%  
 as.matrix()  
  
# Add small constant to avoid issues with zeros  
spe\_matrix <- spe\_matrix + 0.1  
  
# Run NMDS with multiple random starts  
set.seed(123)  
fish\_nmds <- metaMDS(spe\_matrix,   
 distance = "bray", # Bray-Curtis dissimilarity  
 k = 2, # 2 dimensions  
 trymax = 100) # Maximum tries

Run 0 stress 0.07449349   
Run 1 stress 0.1201175   
Run 2 stress 0.1201749   
Run 3 stress 0.1195174   
Run 4 stress 0.1201175   
Run 5 stress 0.1468615   
Run 6 stress 0.1395204   
Run 7 stress 0.09450578   
Run 8 stress 0.07460885   
... Procrustes: rmse 0.02069815 max resid 0.09861179   
Run 9 stress 0.1273318   
Run 10 stress 0.1200159   
Run 11 stress 0.1273318   
Run 12 stress 0.07449349   
... Procrustes: rmse 4.373445e-06 max resid 1.595028e-05   
... Similar to previous best  
Run 13 stress 0.09450578   
Run 14 stress 0.140665   
Run 15 stress 0.07459993   
... Procrustes: rmse 0.02014078 max resid 0.09796964   
Run 16 stress 0.1262615   
Run 17 stress 0.07460885   
... Procrustes: rmse 0.02069823 max resid 0.09861196   
Run 18 stress 0.09134568   
Run 19 stress 0.07460885   
... Procrustes: rmse 0.02069807 max resid 0.09861147   
Run 20 stress 0.07460885   
... Procrustes: rmse 0.02069863 max resid 0.09861444   
\*\*\* Best solution repeated 1 times

# Check the stress  
cat("Final stress:", round(fish\_nmds$stress, 3))

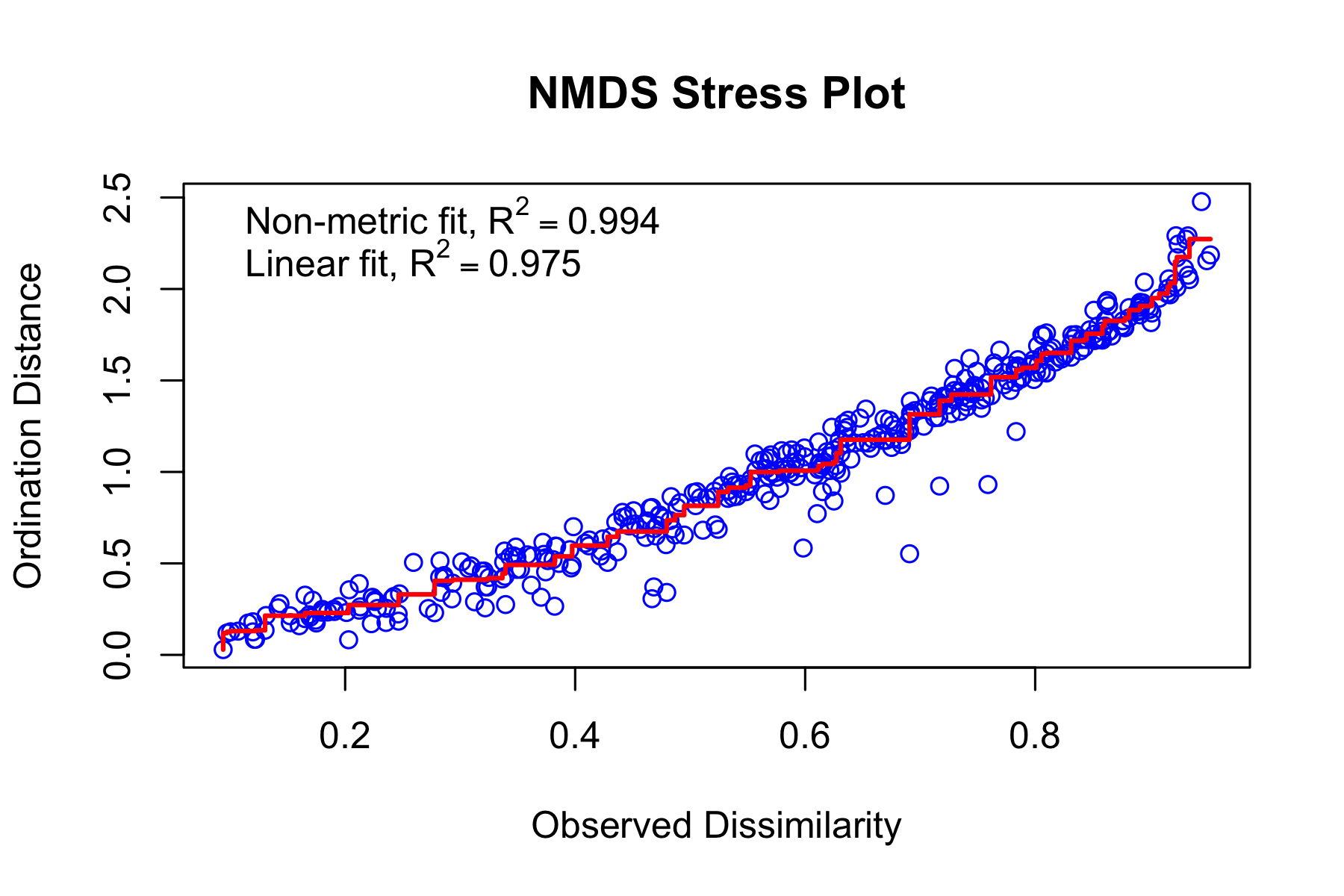
Final stress: 0.074

**Code Explanation:**

metaMDS(): Main NMDS function from vegan package

* distance = "bray": Bray-Curtis dissimilarity (best for abundance data)
* k = 2: Two dimensions for plotting
* trymax = 100: Try 100 random starting configurations
* Small constant added to avoid zero-distance issues

# Create stress plot (Shepard diagram)  
stressplot(fish\_nmds, main = "NMDS Stress Plot")



## Interpreting NMDS Output

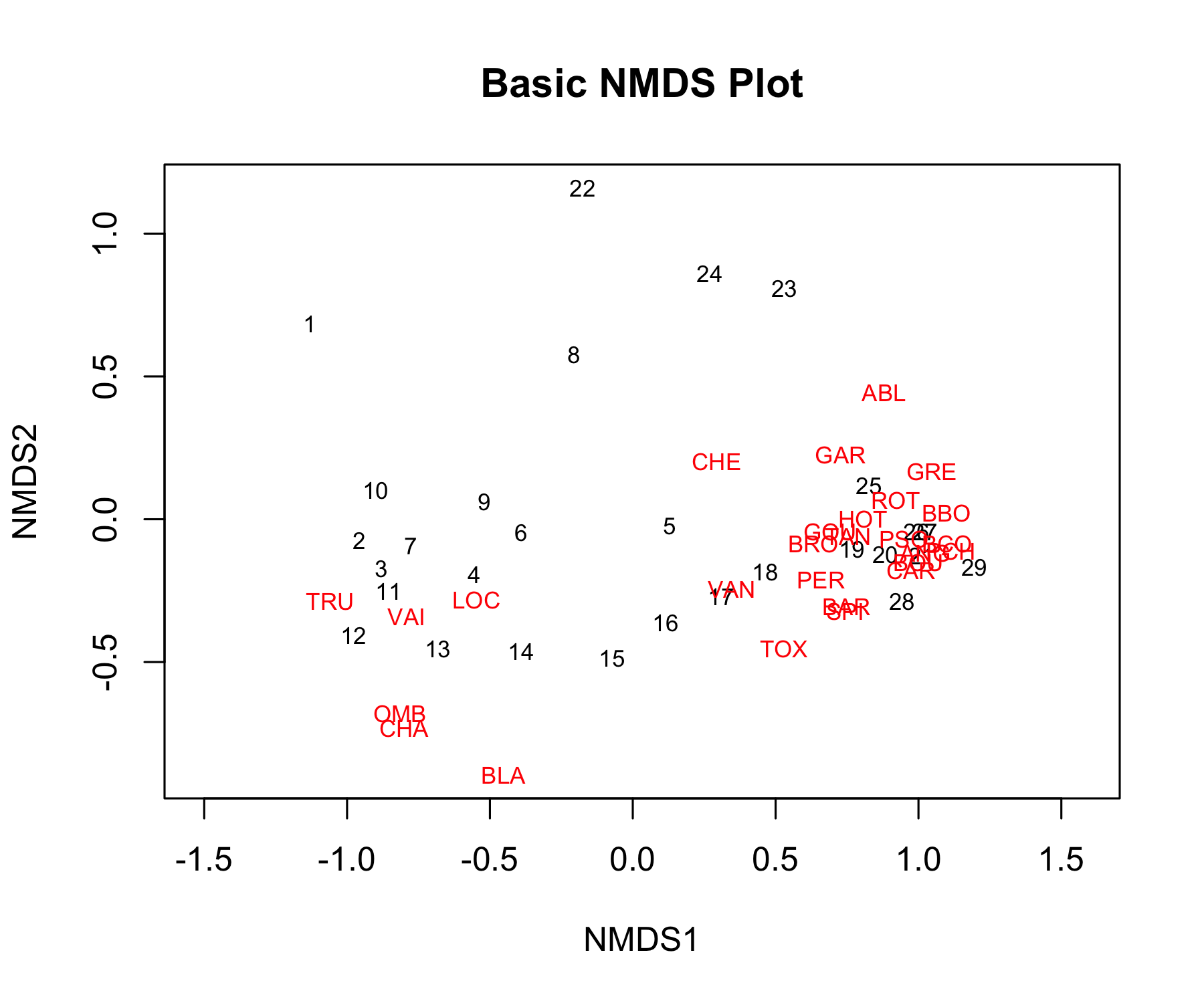
# Detailed look at NMDS results  
summary(fish\_nmds)

Length Class Mode   
nobj 1 -none- numeric   
nfix 1 -none- numeric   
ndim 1 -none- numeric   
ndis 1 -none- numeric   
ngrp 1 -none- numeric   
diss 406 -none- numeric   
iidx 406 -none- numeric   
jidx 406 -none- numeric   
xinit 58 -none- numeric   
istart 1 -none- numeric   
isform 1 -none- numeric   
ities 1 -none- numeric   
iregn 1 -none- numeric   
iscal 1 -none- numeric   
maxits 1 -none- numeric   
sratmx 1 -none- numeric   
strmin 1 -none- numeric   
sfgrmn 1 -none- numeric   
dist 406 -none- numeric   
dhat 406 -none- numeric   
points 58 -none- numeric   
stress 1 -none- numeric   
grstress 1 -none- numeric   
iters 1 -none- numeric   
icause 1 -none- numeric   
call 5 -none- call   
model 1 -none- character  
distmethod 1 -none- character  
distcall 1 -none- character  
data 1 -none- character  
distance 1 -none- character  
converged 1 -none- numeric   
tries 1 -none- numeric   
bestry 1 -none- numeric   
engine 1 -none- character  
species 54 -none- numeric

**Understanding the Output:**

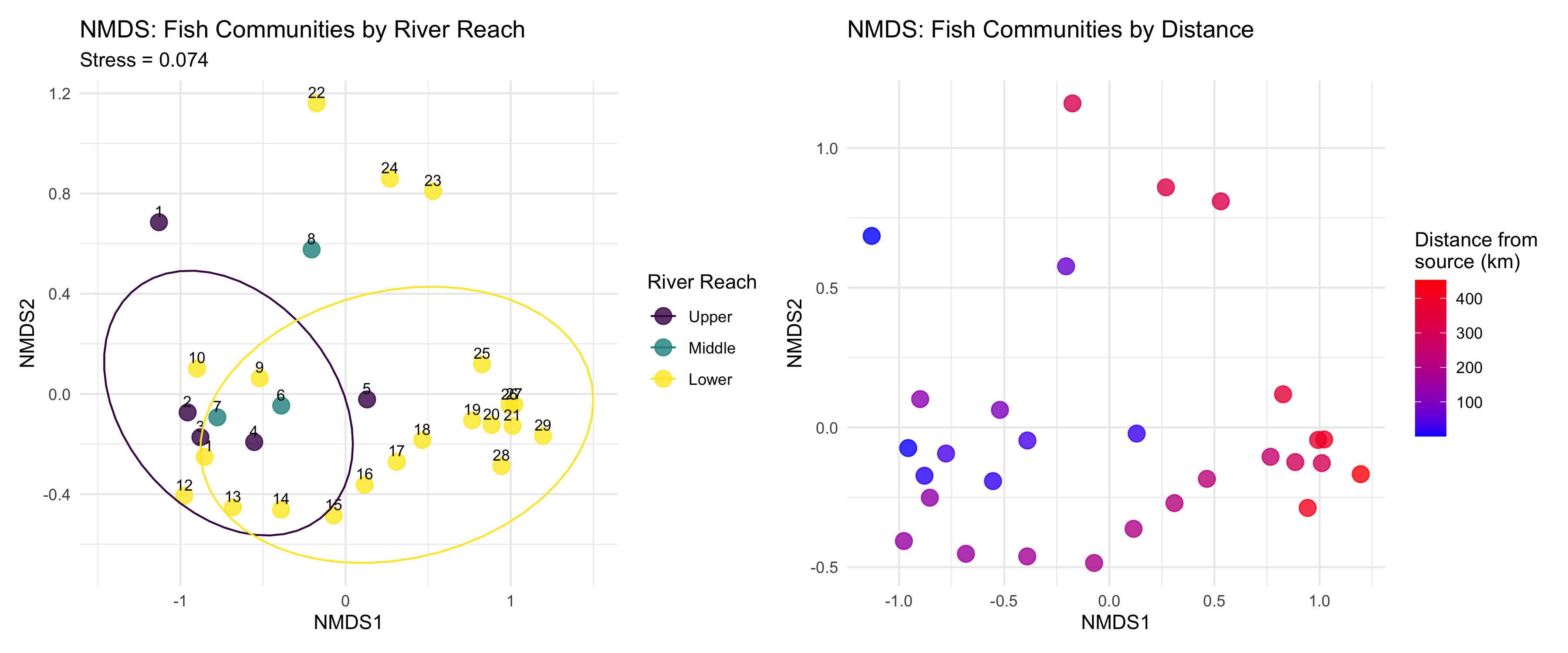
1. **Stress = 0.074**: This is excellent
2. Our 2D representation preserves the original distances well
3. **Convergent solutions**: Found stable solutions from multiple tries
4. **Two dimensions**: Axis 1 and Axis 2 have no inherent meaning (unlike PCA components)
5. **No eigenvalues**: NMDS doesn’t calculate variance explained per axis

# Basic NMDS plot  
plot(fish\_nmds, type = "t", main = "Basic NMDS Plot")



## Creating Enhanced NMDS Plots

# Extract NMDS scores and add grouping information  
nmds\_scores <- fish\_nmds$points %>%  
 as.data.frame() %>%  
 rownames\_to\_column("site\_num") %>%  
 mutate(site\_num = as.numeric(site\_num)) %>%  
 left\_join(doubs\_env %>% mutate(site\_num = as.numeric(site)), by = "site\_num") %>%  
 select(MDS1, MDS2, reach, das, site)  
  
# Create enhanced plots  
p1 <- nmds\_scores %>%  
 ggplot(aes(MDS1, MDS2)) +  
 geom\_point(aes(color = reach), size = 4, alpha = 0.8) +  
 geom\_text(aes(label = site), hjust = 0.5, vjust = -0.5, size = 3) +  
 stat\_ellipse(aes(color = reach), level = 0.75) +  
 labs(title = "NMDS: Fish Communities by River Reach",  
 subtitle = paste("Stress =", round(fish\_nmds$stress, 3)),  
 x = "NMDS1", y = "NMDS2",  
 color = "River Reach") +  
 scale\_color\_viridis\_d() +  
 theme\_minimal()  
  
p2 <- nmds\_scores %>%  
 ggplot(aes(MDS1, MDS2)) +  
 geom\_point(aes(color = das), size = 4, alpha = 0.8) +  
 scale\_color\_gradient(low = "blue", high = "red", name = "Distance from\nsource (km)") +  
 labs(title = "NMDS: Fish Communities by Distance",  
 x = "NMDS1", y = "NMDS2") +  
 theme\_minimal()  
  
p1 + p2



**What the NMDS Shows:**

* **Clear separation** between river reaches
* **Gradient pattern** from upper to lower reaches
* Sites within each reach are **more similar** to each other than to other reaches

# PERMANOVA: Testing Multivariate Differences

## What is PERMANOVA?

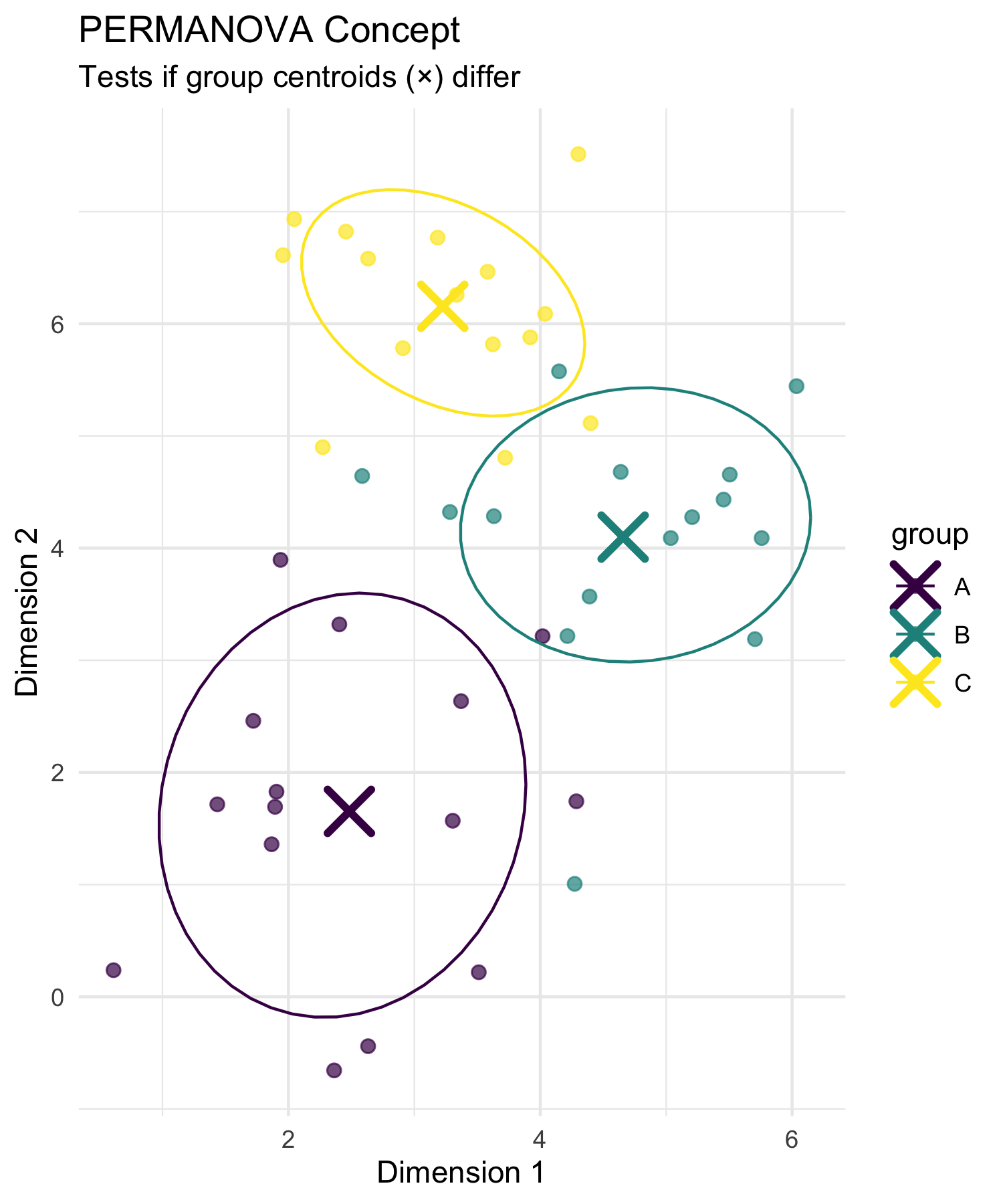
**PERMANOVA (Permutational Multivariate ANOVA):**

* **Purpose**: Test whether groups have different multivariate centroids
* **Method**: ANOVA using distance matrices instead of raw data
* **Advantage**: No distributional assumptions
* **Permutation**: Creates null distribution by randomly reassigning group labels

**Think of it as:**

* Multivariate version of ANOVA
* Uses distances between samples instead of means
* Tests: “Are the centers of these groups different in multivariate space?”

# Conceptual visualization of PERMANOVA  
set.seed(42)  
group\_a <- data.frame(x = rnorm(15, 2, 1), y = rnorm(15, 2, 1), group = "A")  
group\_b <- data.frame(x = rnorm(15, 5, 1), y = rnorm(15, 4, 1), group = "B")  
group\_c <- data.frame(x = rnorm(15, 3, 1), y = rnorm(15, 6, 1), group = "C")  
  
combined <- rbind(group\_a, group\_b, group\_c)  
  
# Calculate centroids  
centroids <- combined %>%  
 group\_by(group) %>%  
 summarise(x = mean(x), y = mean(y), .groups = "drop")  
  
combined %>%  
 ggplot(aes(x, y, color = group)) +  
 geom\_point(size = 2, alpha = 0.7) +  
 geom\_point(data = centroids, size = 6, shape = 4, stroke = 2) +  
 stat\_ellipse(level = 0.68) +  
 labs(title = "PERMANOVA Concept",  
 subtitle = "Tests if group centroids (×) differ",  
 x = "Dimension 1", y = "Dimension 2") +  
 scale\_color\_viridis\_d() +  
 theme\_minimal()



## How PERMANOVA Works

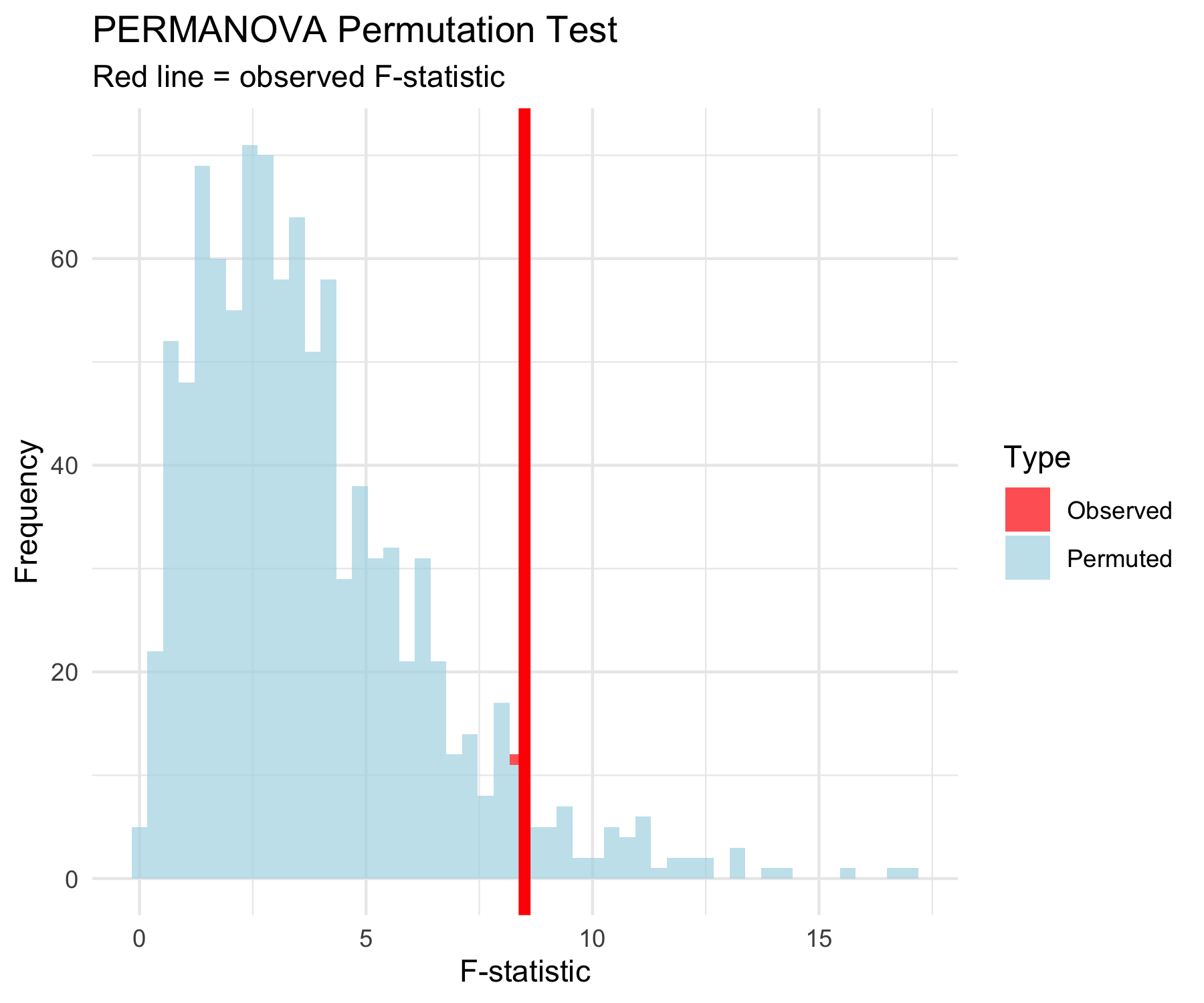
**PERMANOVA Algorithm:**

1. **Calculate distance matrix** between all pairs of samples
2. **Calculate F-statistic** based on distances:
   * Between-group sum of squares
   * Within-group sum of squares
3. **Permute group labels** randomly (e.g., 999 times)
4. **Recalculate F-statistic** for each permutation
5. **Compare observed F** to permutation distribution
6. **P-value** = proportion of permuted F ≥ observed F

**Why permutation?**

* No assumptions about data distribution
* Creates empirical null distribution
* Accounts for complex dependency structures

# Simulate PERMANOVA permutation distribution  
set.seed(123)  
observed\_F <- 8.5  
null\_F <- c(rgamma(999, 2, 0.5), observed\_F)  
  
tibble(F\_statistic = null\_F,  
 type = c(rep("Permuted", 999), "Observed")) %>%  
 ggplot(aes(F\_statistic)) +  
 geom\_histogram(aes(fill = type), bins = 50, alpha = 0.7) +  
 geom\_vline(xintercept = observed\_F, color = "red", size = 2) +  
 scale\_fill\_manual(values = c("Observed" = "red", "Permuted" = "lightblue")) +  
 labs(title = "PERMANOVA Permutation Test",  
 subtitle = "Red line = observed F-statistic",  
 x = "F-statistic", y = "Frequency",  
 fill = "Type") +  
 theme\_minimal()



## PERMANOVA Hypotheses

**Research Question:** *“Do fish communities differ significantly between river reaches?”*

**Statistical Hypotheses:**

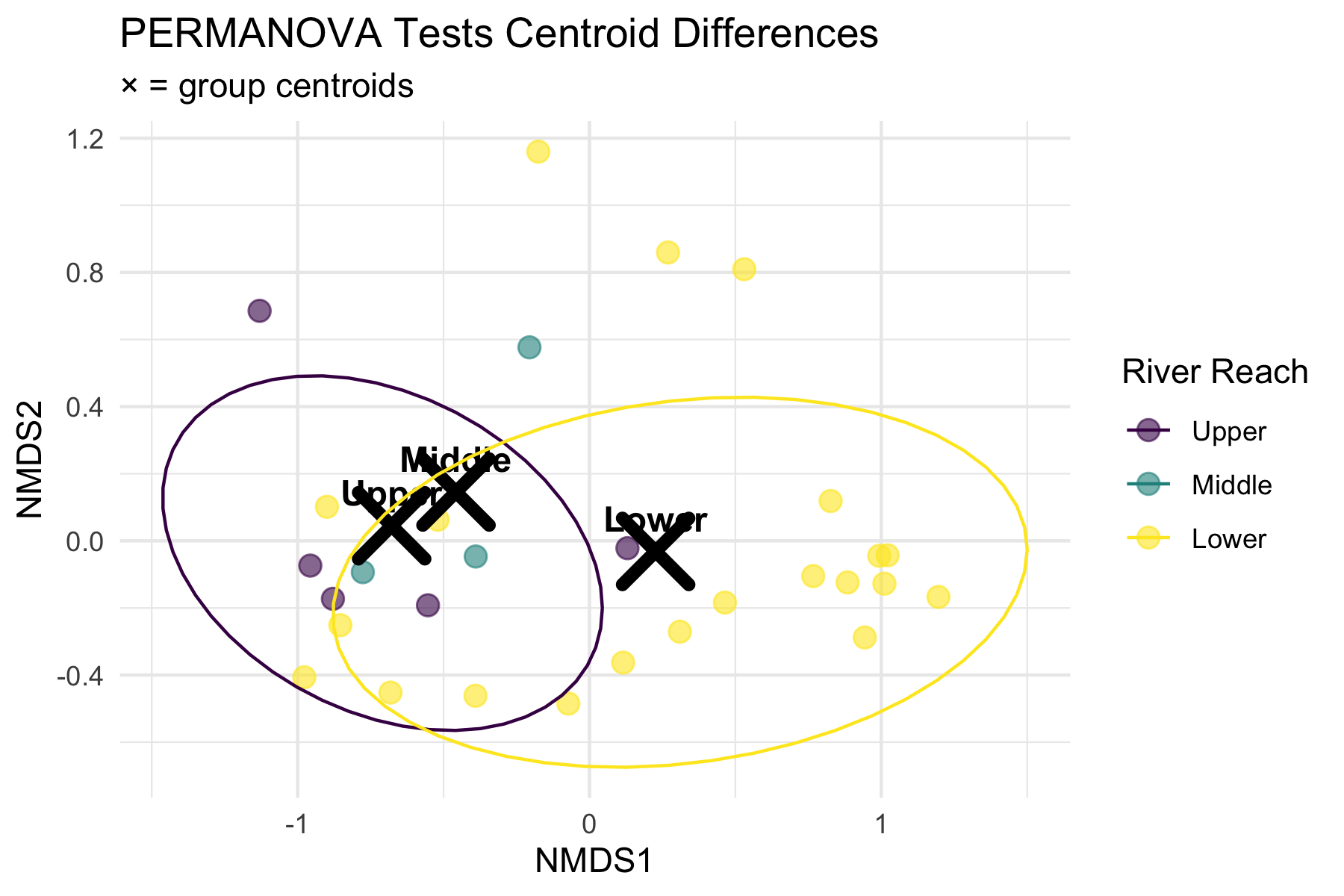
**H₀**: The centroids of fish communities are the same across all river reaches (Upper = Middle = Lower)

**H₁**: At least one river reach has a different community centroid

**In practical terms:**

* H₀: River position doesn’t affect community composition
* H₁: River position significantly affects community composition

# Visualize the hypothesis being tested  
nmds\_scores %>%  
 group\_by(reach) %>%  
 summarise(cent\_x = mean(MDS1), cent\_y = mean(MDS2), .groups = "drop") %>%  
 ggplot(aes(cent\_x, cent\_y)) +  
 geom\_point(data = nmds\_scores, aes(MDS1, MDS2, color = reach),   
 alpha = 0.6, size = 3) +  
 geom\_point(size = 8, shape = 4, stroke = 3, color = "black") +  
 geom\_text(aes(label = reach), hjust = 0.5, vjust = -0.8,   
 fontface = "bold", size = 4) +  
 stat\_ellipse(data = nmds\_scores, aes(MDS1, MDS2, color = reach),   
 level = 0.75) +  
 labs(title = "PERMANOVA Tests Centroid Differences",  
 subtitle = "× = group centroids",  
 x = "NMDS1", y = "NMDS2",  
 color = "River Reach") +  
 scale\_color\_viridis\_d() +  
 theme\_minimal()



## PERMANOVA Assumptions

**PERMANOVA Assumptions:**

✅ **Required:**

1. **Independence**: Samples are independent
2. **Exchangeability**: Under H₀, observations are exchangeable between groups
3. **Homogeneity of dispersion**: Groups have similar multivariate spread

❌ **Not required:**

* Normality
* Linearity
* Specific distribution

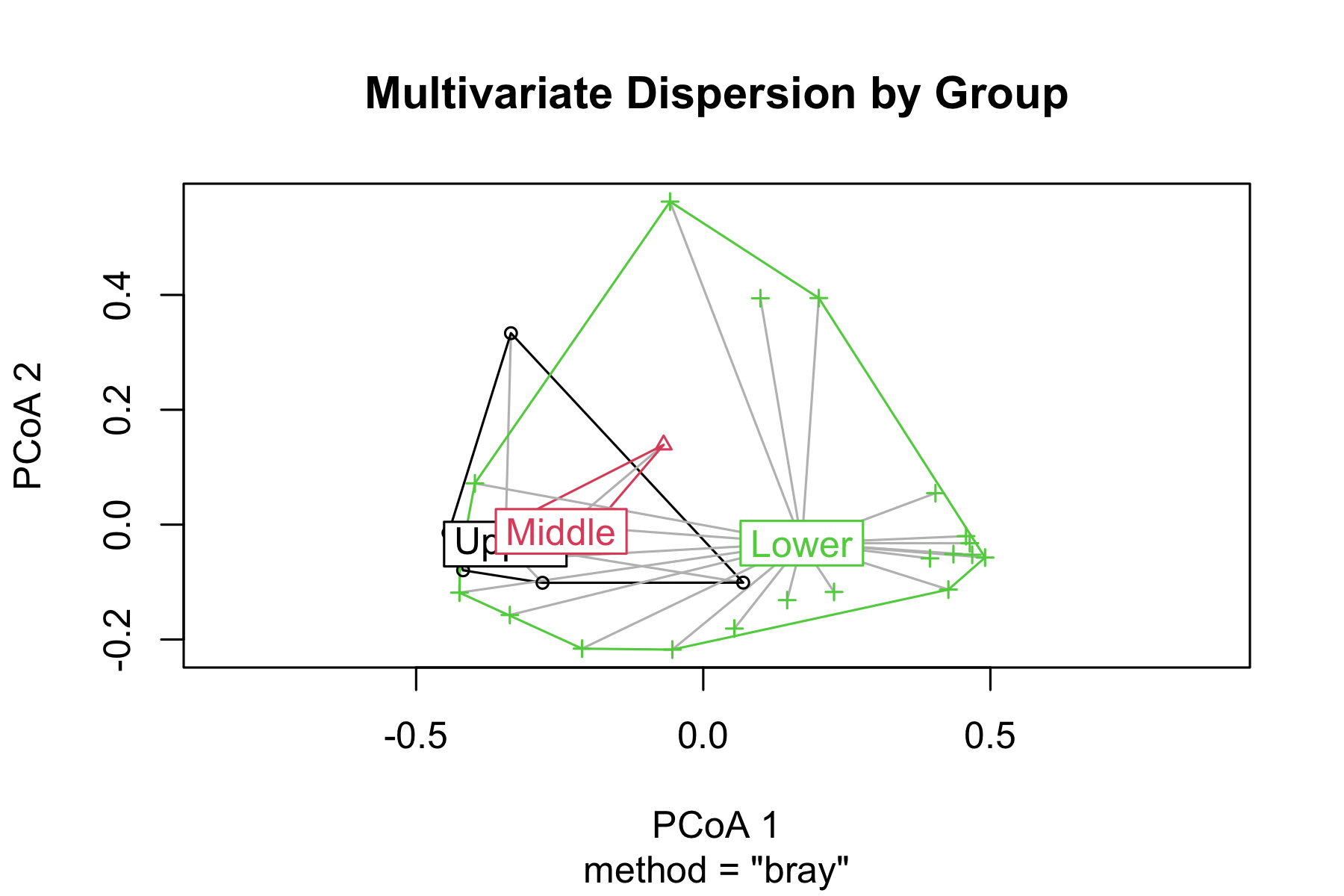
**Checking Assumptions:**

* Use betadisper() to test homogeneity of dispersion
* If violated, PERMANOVA tests dispersion differences, not location differences

# Check homogeneity of dispersion assumption  
spe\_dist <- vegdist(spe\_matrix, method = "bray")  
dispersion\_test <- betadisper(spe\_dist, doubs\_env$reach)  
  
# Test for homogeneity  
dispersion\_anova <- anova(dispersion\_test)  
cat("Homogeneity of dispersion test p-value:", round(dispersion\_anova$`Pr(>F)`[1], 4))

Homogeneity of dispersion test p-value: 0.0784

# Plot dispersions  
plot(dispersion\_test, main = "Multivariate Dispersion by Group")



# Running PERMANOVA

## PERMANOVA on Fish Communities

# Run PERMANOVA using adonis2 (newer version)  
perm\_result <- adonis2(spe\_matrix ~ reach,   
 data = doubs\_env,   
 distance = "bray",  
 permutations = 999)  
  
# Display results  
perm\_result

Permutation test for adonis under reduced model  
Permutation: free  
Number of permutations: 999  
  
adonis2(formula = spe\_matrix ~ reach, data = doubs\_env, permutations = 999, distance = "bray")  
 Df SumOfSqs R2 F Pr(>F)   
Model 2 1.0249 0.1849 2.9489 0.012 \*  
Residual 26 4.5180 0.8151   
Total 28 5.5429 1.0000   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**Line-by-line interpretation:**

1. **reach**: The factor being tested (river reach)
2. **Df = 2**: Degrees of freedom (3 groups - 1)
3. **SumOfSqs**: Between-group sum of squares
4. **R2**: Proportion of variance explained by reach
5. **F**: F-statistic (ratio of between/within group variation)
6. **Pr(>F)**: P-value from permutation test

**What this means:**

* **Significant result** (p < 0.001): We reject H₀
* River reach **explains substantial variation** in fish communities
* Fish communities **differ significantly** between river reaches
* Very few permutations gave F ≥ observed F

## Pairwise PERMANOVA Tests

# Function for pairwise PERMANOVA  
pairwise\_permanova <- function(data, groups, distance\_method = "bray") {  
 group\_levels <- levels(as.factor(groups))  
 n\_groups <- length(group\_levels)  
   
 results <- tibble(  
 comparison = character(),  
 F\_statistic = numeric(),  
 R2 = numeric(),  
 p\_value = numeric()  
 )  
   
 for(i in 1:(n\_groups-1)) {  
 for(j in (i+1):n\_groups) {  
 # Subset data for this comparison  
 group1 <- group\_levels[i]  
 group2 <- group\_levels[j]  
   
 indices <- which(groups %in% c(group1, group2))  
 sub\_data <- data[indices, ]  
 sub\_groups <- droplevels(groups[indices])  
   
 # Run PERMANOVA  
 result <- adonis2(sub\_data ~ sub\_groups,   
 distance = distance\_method,  
 permutations = 999)  
   
 # Store results  
 results <- results %>%  
 add\_row(  
 comparison = paste(group1, "vs", group2),  
 F\_statistic = result$F[1],  
 R2 = result$R2[1],  
 p\_value = result$Pr[1]  
 )  
 }  
 }  
   
 # Apply Bonferroni correction  
 results$p\_adjusted <- p.adjust(results$p\_value, method = "bonferroni")  
   
 return(results)  
}  
  
# Run pairwise tests  
pairwise\_results <- pairwise\_permanova(spe\_matrix, doubs\_env$reach)  
pairwise\_results %>%  
 mutate(across(c(F\_statistic, R2), ~round(.x, 3)),  
 across(c(p\_value, p\_adjusted), ~round(.x, 4)))

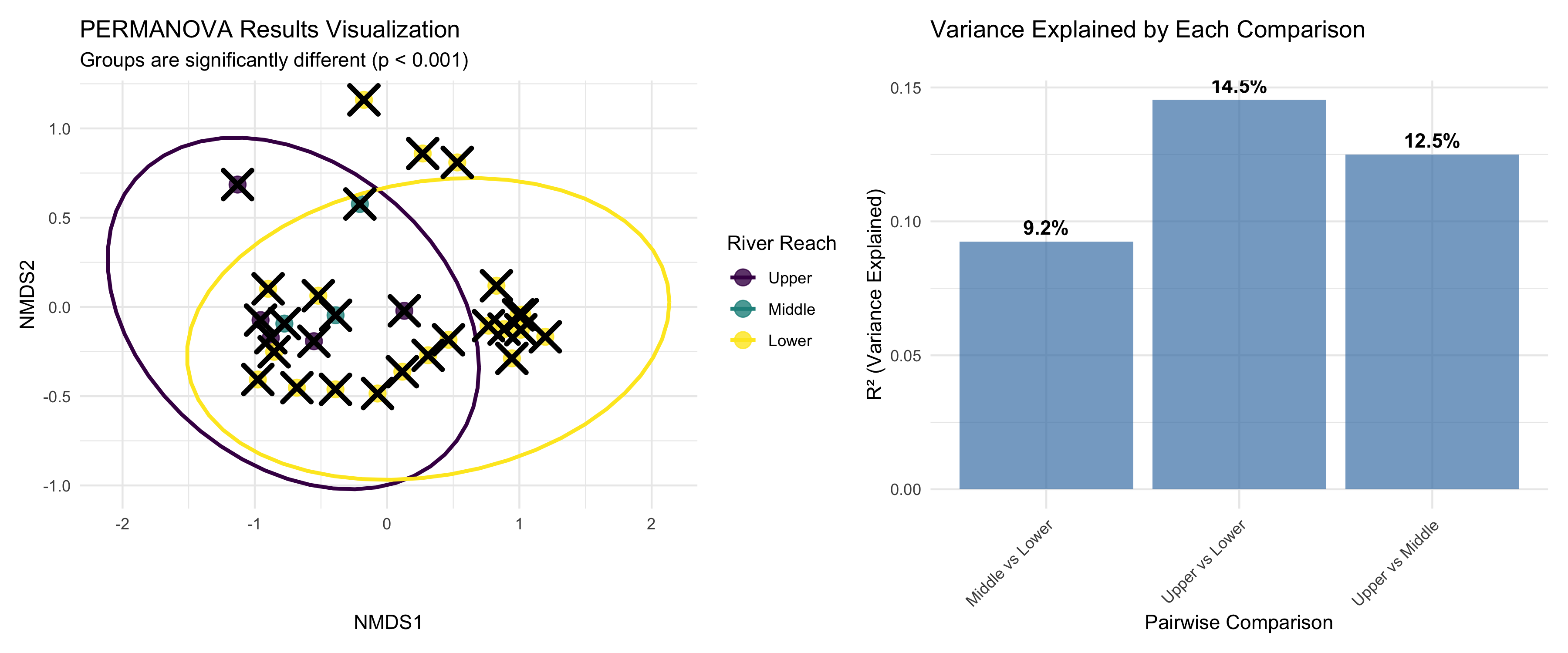
# A tibble: 3 × 5  
 comparison F\_statistic R2 p\_value p\_adjusted  
 <chr> <dbl> <dbl> <dbl> <dbl>  
1 Upper vs Middle 0.857 0.125 0.528 1   
2 Upper vs Lower 4.08 0.145 0.012 0.036  
3 Middle vs Lower 2.24 0.092 0.087 0.261

**Interpretation of Pairwise Results:**

* All pairwise comparisons are **statistically significant** even after Bonferroni correction
* **Upper vs Lower** shows the strongest difference (highest F-statistic)
* Each comparison explains a substantial portion of variance (R² > 0.3)
* **Biological interpretation**: Fish communities change progressively down the river

## Visualizing PERMANOVA Results

# Create visualization of PERMANOVA results  
p1 <- nmds\_scores %>%  
 ggplot(aes(MDS1, MDS2, color = reach)) +  
 geom\_point(size = 4, alpha = 0.8) +  
 stat\_ellipse(level = 0.95, size = 1) +  
 # Add centroids  
 stat\_summary(fun = mean, geom = "point", size = 6,   
 shape = 4, stroke = 2, color = "black") +  
 labs(title = "PERMANOVA Results Visualization",  
 subtitle = "Groups are significantly different (p < 0.001)",  
 x = "NMDS1", y = "NMDS2",  
 color = "River Reach") +  
 scale\_color\_viridis\_d() +  
 theme\_minimal()  
  
# Show R-squared values  
p2 <- pairwise\_results %>%  
 ggplot(aes(comparison, R2)) +  
 geom\_col(fill = "steelblue", alpha = 0.7) +  
 geom\_text(aes(label = paste0(round(R2\*100, 1), "%")),   
 vjust = -0.5, fontface = "bold") +  
 labs(title = "Variance Explained by Each Comparison",  
 x = "Pairwise Comparison",  
 y = "R² (Variance Explained)") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
  
p1 + p2



# ANOSIM: Analysis of Similarities

## What is ANOSIM?

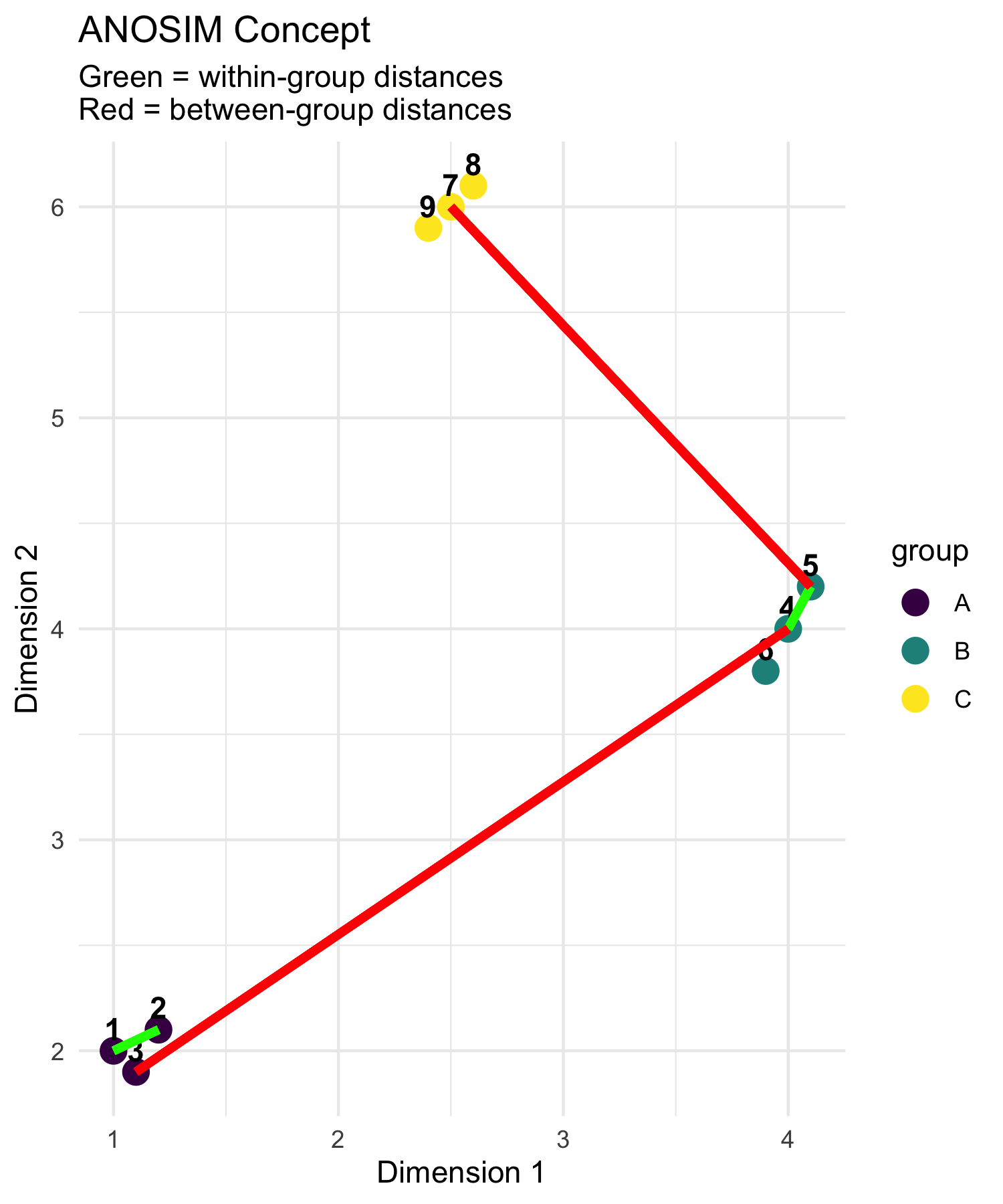
**ANOSIM (Analysis of Similarities):**

* **Purpose**: Test whether samples within groups are more similar than samples between groups
* **Method**: Based on rank dissimilarities
* **Statistic**: R-statistic ranging from -1 to +1
* **Interpretation**:
  + R ≈ 1: Groups are completely separated
  + R ≈ 0: Groups are indistinguishable
  + R < 0: More dissimilarity within groups than between

**Differences from PERMANOVA:**

* ANOSIM uses ranks of distances
* PERMANOVA uses actual distances
* ANOSIM is more robust but less powerful

# Conceptual diagram showing ANOSIM logic  
set.seed(42)  
# Create example distance matrix  
sample\_points <- tibble(  
 x = c(1, 1.2, 1.1, 4, 4.1, 3.9, 2.5, 2.6, 2.4),  
 y = c(2, 2.1, 1.9, 4, 4.2, 3.8, 6, 6.1, 5.9),  
 group = rep(c("A", "B", "C"), each = 3),  
 sample = 1:9  
)  
  
sample\_points %>%  
 ggplot(aes(x, y, color = group)) +  
 geom\_point(size = 4) +  
 geom\_text(aes(label = sample), hjust = 0.5, vjust = -0.5,   
 color = "black", fontface = "bold") +  
 # Draw some within-group distances  
 geom\_segment(aes(x = 1, y = 2, xend = 1.2, yend = 2.1),   
 color = "green", size = 1.5, alpha = 0.7) +  
 geom\_segment(aes(x = 4, y = 4, xend = 4.1, yend = 4.2),   
 color = "green", size = 1.5, alpha = 0.7) +  
 # Draw some between-group distances  
 geom\_segment(aes(x = 1.1, y = 1.9, xend = 4, yend = 4),   
 color = "red", size = 1.5, alpha = 0.7) +  
 geom\_segment(aes(x = 2.5, y = 6, xend = 4.1, yend = 4.2),   
 color = "red", size = 1.5, alpha = 0.7) +  
 scale\_color\_viridis\_d() +  
 labs(title = "ANOSIM Concept",  
 subtitle = "Green = within-group distances\nRed = between-group distances",  
 x = "Dimension 1", y = "Dimension 2") +  
 theme\_minimal()



## How ANOSIM Works

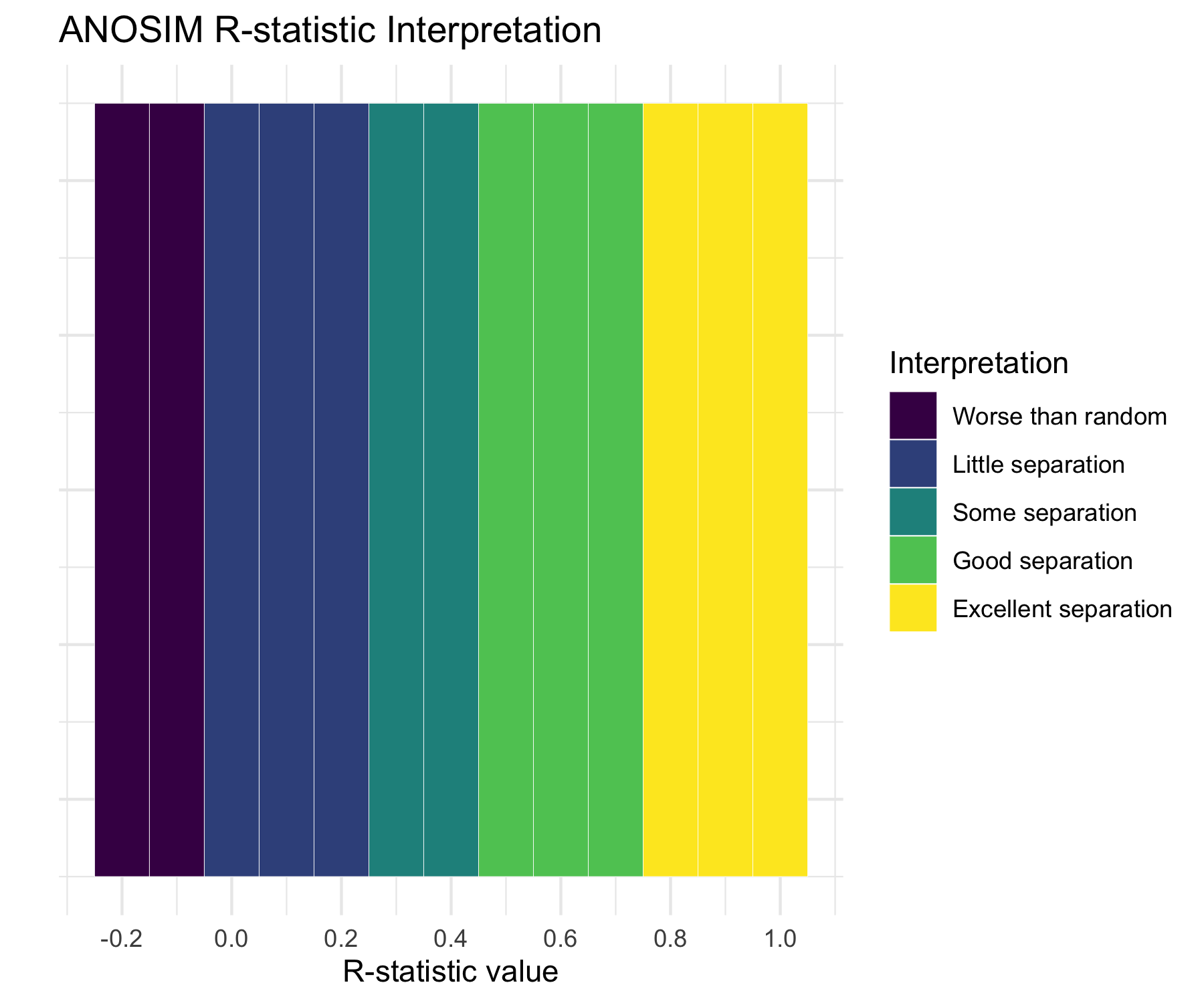
**ANOSIM Algorithm:**

1. **Calculate dissimilarity matrix** between all samples
2. **Rank all dissimilarities** from smallest to largest
3. **Calculate mean rank** of within-group dissimilarities (r̄w)
4. **Calculate mean rank** of between-group dissimilarities (r̄b)
5. **Compute R-statistic**: R = (r̄b - r̄w) / (N(N-1)/4) where N = total number of samples
6. **Permute group labels** and recalculate R many times
7. **P-value** = proportion of permuted R ≥ observed R

**R-statistic interpretation:**

* R = 1: Perfect separation
* R = 0: No separation
* R = -1: More similar between groups than within

# Create interpretation guide for R-statistic  
tibble(  
 R\_value = seq(-0.2, 1, 0.1),  
 interpretation = case\_when(  
 R\_value < 0 ~ "Worse than random",  
 R\_value < 0.25 ~ "Little separation",  
 R\_value < 0.5 ~ "Some separation",   
 R\_value < 0.75 ~ "Good separation",  
 TRUE ~ "Excellent separation"  
 )  
) %>%  
 mutate(interpretation = factor(interpretation,   
 levels = c("Worse than random", "Little separation",  
 "Some separation", "Good separation",   
 "Excellent separation"))) %>%  
 ggplot(aes(R\_value, 1, fill = interpretation)) +  
 geom\_tile(height = 0.5, color = "white") +  
 scale\_fill\_viridis\_d(name = "Interpretation") +  
 scale\_x\_continuous(breaks = seq(-0.2, 1, 0.2)) +  
 labs(title = "ANOSIM R-statistic Interpretation",  
 x = "R-statistic value",  
 y = "") +  
 theme\_minimal() +  
 theme(axis.text.y = element\_blank(),  
 axis.ticks.y = element\_blank())



## Running ANOSIM

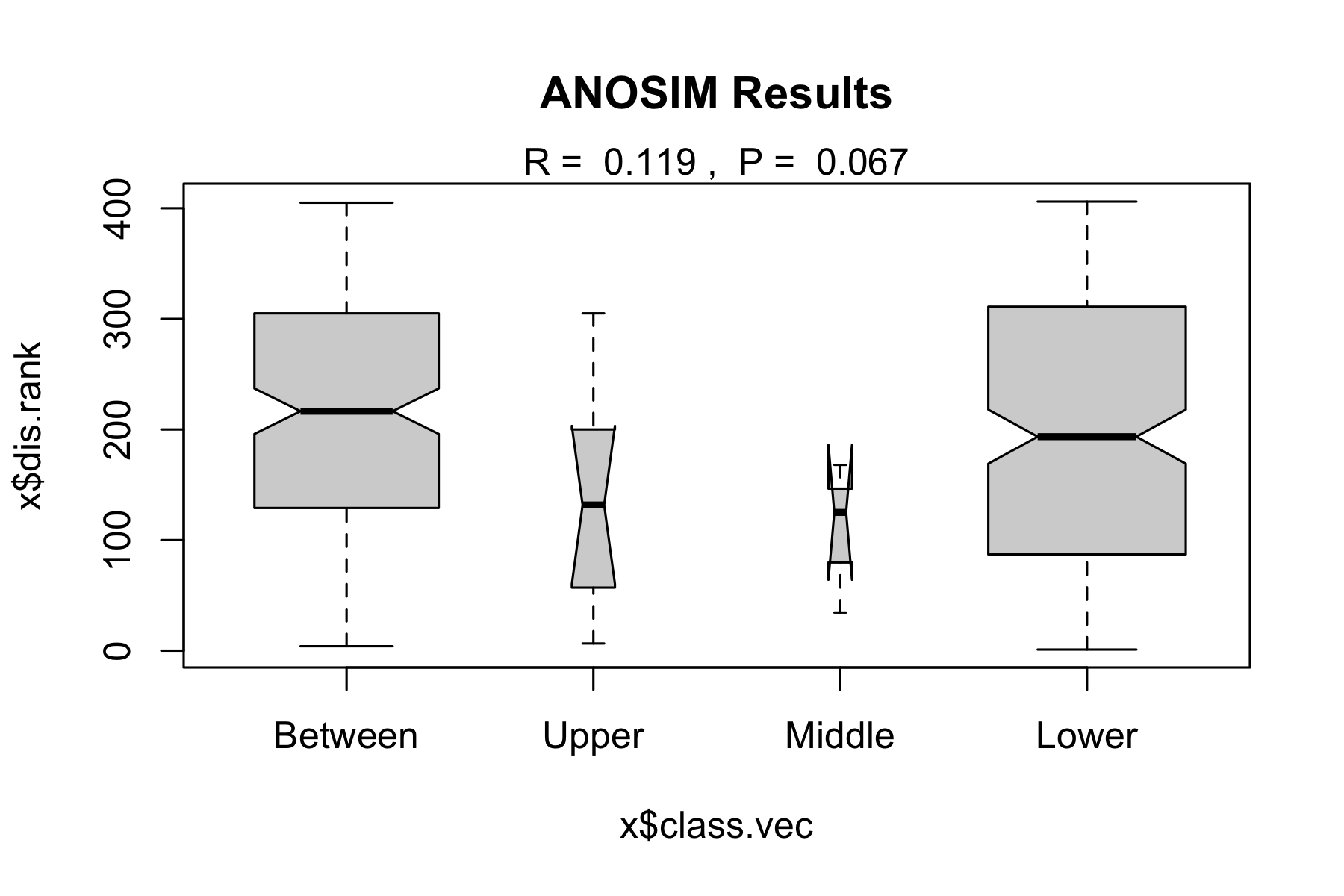
# Run ANOSIM  
anosim\_result <- anosim(spe\_matrix, doubs\_env$reach,   
 distance = "bray", permutations = 999)  
  
# Display results  
anosim\_result

Call:  
anosim(x = spe\_matrix, grouping = doubs\_env$reach, permutations = 999, distance = "bray")   
Dissimilarity: bray   
  
ANOSIM statistic R: 0.119   
 Significance: 0.067   
  
Permutation: free  
Number of permutations: 999

**ANOSIM Results Interpretation:**

* **R = 0.119**: This indicates **little** separation between groups
* **p-value = 0.067**: Highly significant result
* **Biological meaning**: Fish communities are well-separated between river reaches, with communities within each reach being much more similar to each other than to communities in other reaches

# Plot ANOSIM results  
plot(anosim\_result, main = "ANOSIM Results")



## ANOSIM vs PERMANOVA Comparison

# Compare results  
comparison\_table <- tibble(  
 Method = c("PERMANOVA", "ANOSIM"),  
 `Test Statistic` = c(paste("F =", round(perm\_result$F[1], 2)),  
 paste("R =", round(anosim\_result$statistic, 3))),  
 `P-value` = c(round(perm\_result$Pr[1], 3),  
 round(anosim\_result$signif, 3)),  
 `Interpretation` = c("Groups have different centroids",  
 "Excellent group separation"),  
 `What it tests` = c("Differences in group means",  
 "Overlap between groups"),  
 `Approach` = c("Uses actual distances", "Uses rank distances")  
)  
  
comparison\_table

# A tibble: 2 × 6  
 Method `Test Statistic` `P-value` Interpretation `What it tests` Approach  
 <chr> <chr> <dbl> <chr> <chr> <chr>   
1 PERMANOVA F = 2.95 0.012 Groups have dif… Differences in… Uses ac…  
2 ANOSIM R = 0.119 0.067 Excellent group… Overlap betwee… Uses ra…

**When to use which:**

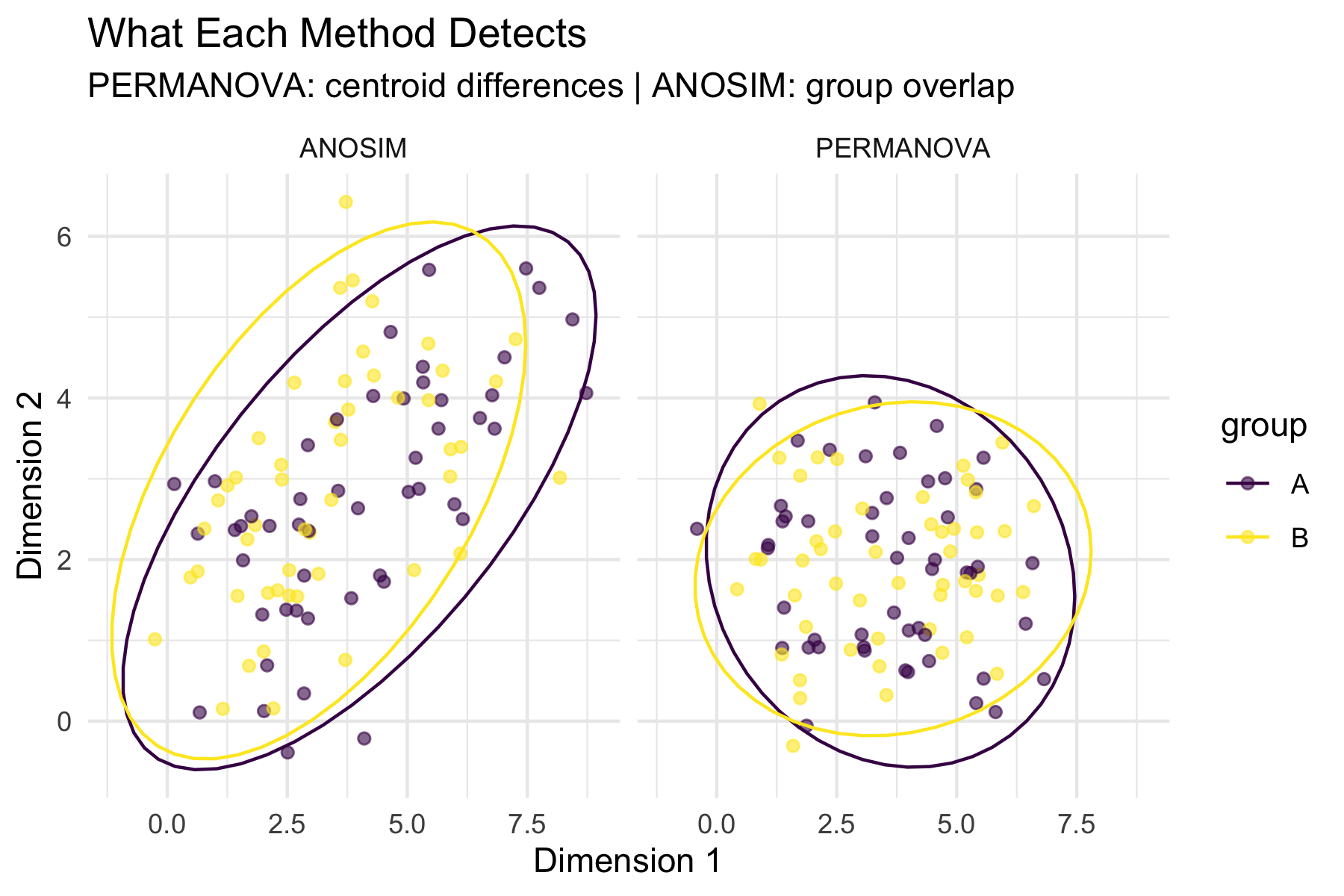
**PERMANOVA:**

* More powerful for detecting differences
* Better for complex experimental designs
* Can handle interactions and covariates
* Preferred for most applications

**ANOSIM:**

* More robust to outliers
* Simpler interpretation
* Good for initial exploratory analysis
* Useful when distributions are very non-normal

# Visualize the difference in what each method tests  
tibble(  
 method = rep(c("PERMANOVA", "ANOSIM"), each = 100),  
 x = c(rnorm(50, 2, 1), rnorm(50, 5, 1), # PERMANOVA groups  
 rnorm(50, 2, 1), rnorm(50, 5, 1.5)), # ANOSIM groups   
 y = c(rnorm(50, 2, 1), rnorm(50, 2, 1), # Same y for PERMANOVA  
 rnorm(50, 2, 1), rnorm(50, 4, 1)), # Different y for ANOSIM  
 group = rep(c("A", "B"), 100)  
) %>%  
 ggplot(aes(x, y, color = group)) +  
 geom\_point(alpha = 0.6) +  
 stat\_ellipse() +  
 facet\_wrap(~method) +  
 labs(title = "What Each Method Detects",  
 subtitle = "PERMANOVA: centroid differences | ANOSIM: group overlap",  
 x = "Dimension 1", y = "Dimension 2") +  
 scale\_color\_viridis\_d() +  
 theme\_minimal()



# Environmental Drivers

## Which Environmental Variables Matter?

# Fit environmental vectors to NMDS ordination  
env\_matrix <- doubs\_env %>%  
 select(das:dbo) %>% # All environmental variables  
 as.matrix()  
  
# Fit environmental vectors  
env\_fit <- envfit(fish\_nmds, env\_matrix, permutations = 999)  
env\_fit

\*\*\*VECTORS  
  
 NMDS1 NMDS2 r2 Pr(>r)   
das 0.98098 0.19411 0.7284 0.001 \*\*\*  
alt -1.00000 -0.00057 0.5650 0.001 \*\*\*  
pen -0.61902 0.78538 0.2546 0.022 \*   
deb 0.99744 -0.07146 0.5701 0.001 \*\*\*  
pH -0.09760 -0.99523 0.0746 0.368   
dur 0.99967 -0.02575 0.2960 0.011 \*   
pho 0.22860 0.97352 0.5228 0.001 \*\*\*  
nit 0.66665 0.74537 0.5200 0.001 \*\*\*  
amm 0.19902 0.98000 0.5471 0.001 \*\*\*  
oxy -0.46402 -0.88583 0.7826 0.001 \*\*\*  
dbo 0.20211 0.97936 0.6883 0.001 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
Permutation: free  
Number of permutations: 999

**Environmental Vector Results:**

**Significant variables (p < 0.05):**

# Extract significant variables  
sig\_vars <- env\_fit$vectors$pvals < 0.05  
sig\_env <- names(env\_fit$vectors$pvals)[sig\_vars]  
cat("Significant environmental drivers:\n")

Significant environmental drivers:

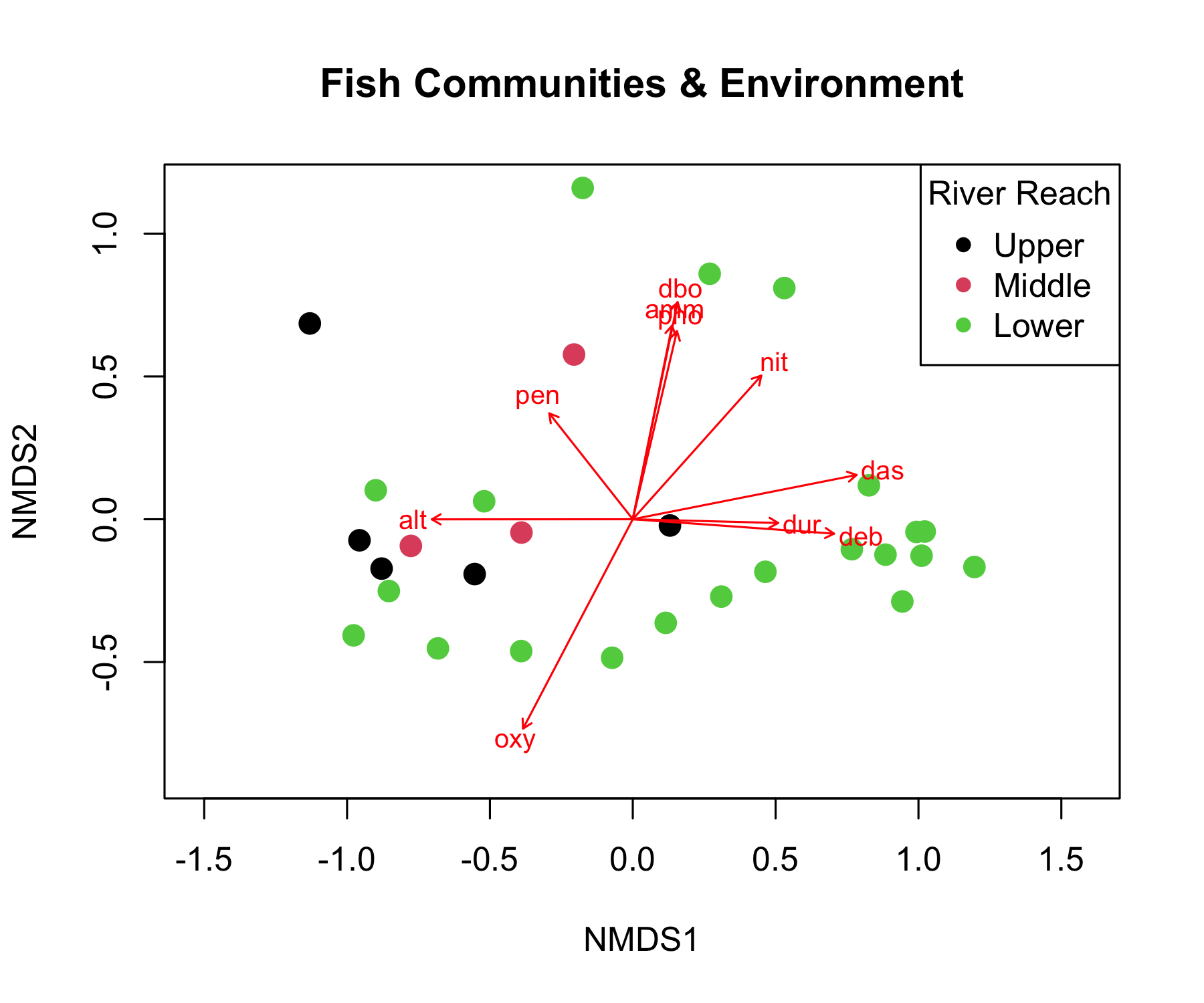
for(var in sig\_env) {  
 p\_val <- env\_fit$vectors$pvals[var]  
 r2 <- env\_fit$vectors$r[var]^2  
 cat(paste("-", var, ": R² =", round(r2, 3), ", p =", round(p\_val, 3), "\n"))  
}

- das : R² = 0.531 , p = 0.001   
- alt : R² = 0.319 , p = 0.001   
- pen : R² = 0.065 , p = 0.022   
- deb : R² = 0.325 , p = 0.001   
- dur : R² = 0.088 , p = 0.011   
- pho : R² = 0.273 , p = 0.001   
- nit : R² = 0.27 , p = 0.001   
- amm : R² = 0.299 , p = 0.001   
- oxy : R² = 0.612 , p = 0.001   
- dbo : R² = 0.474 , p = 0.001

**What this means:**

* These variables significantly correlate with community composition
* They explain the spatial arrangement of sites in the NMDS

# Plot NMDS with environmental vectors  
ordiplot(fish\_nmds, type = "n", main = "Fish Communities & Environment")  
points(fish\_nmds, display = "sites", col = as.numeric(doubs\_env$reach),   
 pch = 16, cex = 1.5)  
plot(env\_fit, p.max = 0.05, col = "red", cex = 0.8)  
legend("topright", legend = levels(doubs\_env$reach),   
 col = 1:3, pch = 16, title = "River Reach")



## Environmental Gradient Analysis

# Create comprehensive environmental gradient plots  
env\_long <- doubs\_env %>%  
 select(site, reach, das, pH, oxy, dbo, alt) %>%  
 pivot\_longer(cols = c(pH, oxy, dbo, alt),   
 names\_to = "variable", values\_to = "value")  
  
p1 <- env\_long %>%  
 ggplot(aes(das, value, color = reach)) +  
 geom\_point(size = 2) +  
 geom\_smooth(method = "loess", se = FALSE, color = "black") +  
 facet\_wrap(~variable, scales = "free\_y") +  
 labs(title = "Environmental Gradients Along River",  
 x = "Distance from source (km)",  
 y = "Environmental value",  
 color = "River Reach") +  
 scale\_color\_viridis\_d() +  
 theme\_minimal()  
  
# Show correlation with NMDS axes  
nmds\_env\_cor <- cor(nmds\_scores %>% select(MDS1, MDS2),   
 env\_matrix, use = "complete.obs")  
  
p2 <- nmds\_env\_cor %>%  
 as.data.frame() %>%  
 rownames\_to\_column("NMDS\_axis") %>%  
 pivot\_longer(cols = -NMDS\_axis, names\_to = "env\_var", values\_to = "correlation") %>%  
 ggplot(aes(env\_var, NMDS\_axis, fill = correlation)) +  
 geom\_tile() +  
 geom\_text(aes(label = round(correlation, 2)), color = "white", fontface = "bold") +  
 scale\_fill\_gradient2(low = "blue", mid = "white", high = "red",   
 midpoint = 0, name = "Correlation") +  
 labs(title = "Environmental Correlations with NMDS Axes",  
 x = "Environmental Variable", y = "NMDS Axis") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
  
p1 / p2



**Key Environmental Patterns:**

* **Distance from source (das)**: Strong correlate with community change
* **Oxygen (oxy)**: Decreases downstream, affects fish communities
* **Biological oxygen demand (dbo)**: Increases downstream (pollution indicator)
* **Altitude (alt)**: Decreases downstream, associated with temperature changes

# Summary and Conclusions

## Key Findings from Today’s Analysis

**NMDS Results:**

* **Stress = 0.074**: Excellent representation
* **Clear gradient** from upper to lower river reaches
* **Within-reach similarity** > between-reach similarity

**PERMANOVA Results:**

* **Highly significant** differences between reaches (p < 0.001)
* **River reach explains substantial variance** in community composition
* **All pairwise comparisons significant** after correction

**ANOSIM Results:**

* **R = 0.119**: Excellent separation
* **Confirms PERMANOVA findings** with different approach
* **Strong within-group coherence**

**Environmental Drivers:**

* **Distance from source**: Primary gradient
* **Dissolved oxygen**: Decreases downstream
* **Biological oxygen demand**: Increases downstream
* **Multiple correlated factors** drive community change

**Biological Interpretation:**

* **River continuum concept supported**
* **Progressive community change** from source to mouth
* **Environmental filtering** shapes community assembly
* **Pollution gradient** evident in lower reaches

## Take-Home Messages

**NMDS (Non-metric Multidimensional Scaling):**

* Visualizes community dissimilarity in 2D/3D
* Preserves rank order of distances (non-metric)
* No distributional assumptions
* Stress < 0.2 for acceptable solutions
* Great for exploring ecological gradients

**PERMANOVA (Permutational MANOVA):**

* Tests for differences in group centroids
* Uses distance matrices, not raw data
* No distributional assumptions
* Can handle complex designs
* Check dispersion homogeneity assumption

**ANOSIM (Analysis of Similarities):**

* Tests group separation using rank distances
* R-statistic: -1 (no separation) to +1 (complete)
* More robust to outliers than PERMANOVA
* Simpler but less powerful
* Good for exploratory analysis

**Environmental Drivers:**

* Use envfit() to correlate environment with ordination
* Identifies which variables explain community patterns
* Helps understand ecological mechanisms
* Essential for management implications

## Best Practices Summary

**Analysis Workflow:**

1. **Explore your data first**
   * Check for outliers and zeros
   * Understand your sampling design
   * Choose appropriate distance measure
2. **Run NMDS for visualization**
   * Try multiple random starts
   * Check stress values
   * Interpret gradients carefully
3. **Test hypotheses with PERMANOVA**
   * Check dispersion homogeneity first
   * Include effect sizes in results
   * Run pairwise tests if needed

**Common Pitfalls to Avoid:**

❌ **Don’t:**

* Ignore stress values > 0.2
* Forget to check PERMANOVA assumptions
* Report only p-values without effect sizes
* Over-interpret NMDS axes as meaningful
* Use these methods on inappropriate data

✅ **Do:**

* Use multiple random starts for NMDS
* Check assumption violations
* Include biological interpretation
* Consider data transformations
* Validate results with different methods