Overlapping polygons

- Function `chull` gets points for polygons
- Function `joinPolys` (PBSmapping package) creates a polygon describing overlap
- Function `TT.polygon.area` (soiltexture package) calculates areas

```r
> TT.polygon.area(p1_hull$X, p1_hull$Y)
```

```r
[1] 13.18379
```

```r
> TT.polygon.area(p2_hull$X, p2_hull$Y)
```

```r
[1] 9.355743
```

```r
> TT.polygon.area(p3$X, p3$Y)
```

```r
[1] 0.3174184
```

Summarizing output from ordination

- Function `scores` will return scaled ordination scores from any ordination object in vegan.

```r
prin_comp <- rda(env_data[, 2:14], scale = TRUE)
scores(prin_comp)
```

- Use choices option to specify which axes

```r
scores(prin_comp, choices = c(3, 4))
```

Remember to standardize your data for PCA

- Remember that PCA is simply trying to capture and summarize variability in the data you give it.
- Variance in the environmental dataset:
### PCA Centroids

- Use **`tapply`** function to calculate centroids for groups of points
  
  ```r
drain_cent_axis1 <- tapply(pca_scores$sites[,1], env_data$drainage, mean)
drain_cent_axis2 <- tapply(pca_scores$sites[,2], env_data$drainage, mean)
```

### Interpretation of loadings

<table>
<thead>
<tr>
<th>Variable</th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
<th>PC5</th>
<th>PC6</th>
</tr>
</thead>
<tbody>
<tr>
<td>drainage_area</td>
<td>1.2341</td>
<td>0.2776</td>
<td>-0.205</td>
<td>0.0088</td>
<td>-0.0198</td>
<td>0.098</td>
</tr>
<tr>
<td>pH</td>
<td>-0.1078</td>
<td>-0.8974</td>
<td>-0.4836</td>
<td>0.7306</td>
<td>-0.1004</td>
<td>-0.2323</td>
</tr>
<tr>
<td>NO</td>
<td>-0.6685</td>
<td>-0.1879</td>
<td>-0.2584</td>
<td>-0.1669</td>
<td>0.4505</td>
<td>-0.3522</td>
</tr>
<tr>
<td>Temp</td>
<td>-0.9963</td>
<td>-0.1767</td>
<td>-0.9328</td>
<td>-0.9875</td>
<td>-0.2309</td>
<td>0.4903</td>
</tr>
<tr>
<td>tds</td>
<td>-0.2297</td>
<td>-0.7724</td>
<td>-0.3014</td>
<td>0.2034</td>
<td>-0.9136</td>
<td>-0.196</td>
</tr>
<tr>
<td>DO</td>
<td>-0.5177</td>
<td>-0.9427</td>
<td>-0.4806</td>
<td>0.1552</td>
<td>0.1921</td>
<td>0.0521</td>
</tr>
<tr>
<td>cover</td>
<td>0.9686</td>
<td>-0.4837</td>
<td>0.4836</td>
<td>0.7237</td>
<td>0.8706</td>
<td>0.4333</td>
</tr>
<tr>
<td>veg</td>
<td>0.8976</td>
<td>-0.2266</td>
<td>-0.7984</td>
<td>-0.5754</td>
<td>0.2225</td>
<td>0.6404</td>
</tr>
<tr>
<td>substrate</td>
<td>0.1489</td>
<td>-0.4927</td>
<td>-0.7789</td>
<td>-0.3052</td>
<td>-0.1005</td>
<td>-0.7267</td>
</tr>
<tr>
<td>TSS</td>
<td>-0.4836</td>
<td>-0.2584</td>
<td>-0.1669</td>
<td>0.4505</td>
<td>-0.3522</td>
<td>0.6518</td>
</tr>
<tr>
<td>canopy</td>
<td>0.5186</td>
<td>0.3601</td>
<td>-0.4609</td>
<td>0.3040</td>
<td>0.2134</td>
<td>0.8706</td>
</tr>
<tr>
<td>turbidity</td>
<td>0.1470</td>
<td>0.9125</td>
<td>-0.1919</td>
<td>0.6162</td>
<td>0.1082</td>
<td>-0.3256</td>
</tr>
</tbody>
</table>

### Non Metric Multidimensional Scaling (NMDS)

- Most robust unconstrained ordination for community data
- Distance based
- Similar to metric multidimensional scaling (PCoA)
  - rank distances used instead (thus non metric)
  - Use of ranks aligns it well with ANOSIM
- Usually called NMDS, sometimes called MDS
- Iterative, not eigen based

### Table 9.1 from Legendre and Legendre (1998)

<table>
<thead>
<tr>
<th>Method</th>
<th>Distance</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>Euclidean</td>
<td>Quantitative, linear relationships assumed, beware of double-zeros</td>
</tr>
<tr>
<td>PCoA</td>
<td>Any</td>
<td>Quantitative, qualitative or mixed</td>
</tr>
<tr>
<td>NMDS</td>
<td>Any</td>
<td>Quantitative, qualitative or mixed</td>
</tr>
<tr>
<td>CA</td>
<td>X²</td>
<td>Non-negative, quantitative or binary</td>
</tr>
<tr>
<td>FA</td>
<td>Euclidean</td>
<td>Quantitative, linear relationships assumed, beware of double-zeros</td>
</tr>
</tbody>
</table>

### NMDS

- **Goal of analysis** – place samples in k dimensional space to minimize differences between **rank similarities** in the original distance matrix and **rank** euclidean similarities in ordination space.
**NDMS**
- As with PCoA:
  - Need to specify number of axes
- "quality" of representation improves with higher k, perfect representation when k=number of variables
- The goal is to simplify, so typically want 2 or 3 axes
- No eigen vectors and values, direct links to variables is lost (need to use weighted average scores)

**NMDS**
- Various measures of stress.
  - Simple correlation between original distance matrix and distance in ordination space.
  - Recall that NMDS is non-metric and tried to match rank similarities. Other measures of stress involve residuals from monotonic regressions.
  - Note that some functions report this as a percentage and some as a proportion.
  - Function `stressplot()` can be used with `nmds` results to visualize.

**NMDS**
- Stress – measure of the lack of fit between rank order dissimilarities and rank order euclidean distance in ordination space.
  - Analogous to cophenetic correlation or other similar measures
  - Unbound but expressed as a scaled percentage.
  - 0-100%, higher = worse fit
  - Generally, <15% is good
  - Stress always reduced by adding a dimension, but the goal is to use as few dimensions as possible.

**NMDS**
- Species information lost (distance based)
  - No % variation accounted for by axes, stress is an analog and stress can be partitioned by axis
  - No strict order of importance to axes
  - Points and axes can be fully rotated and scaled, all that matters is their relative position in k-dimensional space
  - Must specify number of axes ahead of time
    - Stress is reduced as more axes are used. More axes = less information contained on each.
    - 1st dimension of a 2D NMDS is not the same thing as the 1st dimension of a 6D NMDS
    - This is very different from PCoA
  - Iterative process, consecutive runs may differ
  - Computationally intensive
Example NMDS from the literature.
Axes are unlabeled, without units
and no % variance accounted for.

NMDS
- Starts with initial configuration (solution), moves points each iteration to reduce stress (seeks to converge on a best solution)
- Because it is iterative, NMDS is
  - Computationally intensive
  - Susceptible to getting stuck in local optima
  - Typically use multiple runs with different starting points

Multidimensional Space

isoMDS Code
- Plain NMDS
  - Code (MASS package):
    - distance<-vegdist(community, method="bray")
    - nmds<-isoMDS(distance, k=2)
  - ordiplot(nmds)
- Options
  - K – number of axes
  - Tol – convergence tolerance
  - Maxit – maximum number of iterations
  - Can specify a starting configuration, otherwise will perform a PCoA to obtain a starting position
- Output
  - Stress at each iteration
  - Final stress
  - Point scores for samples in k dimensions
  - Can be used with missing data
  - Stressplot function does not work

Convergence on a solution
- isoMDS converges on a configuration that minimizes stress, stops when additional iterations do not improve stress

larger tolerance = convergence reached earlier with fewer iterations.
Tradeoff in computing time.

- Note that isoMDS uses one starting configuration (susceptible to local optima problem)
If you want a good 2D representation, k=2 is better than k=10 even though the stress will be higher.

**Stress Plots**

- **K=10**
- **K=2, first 2 axes**

**Stress Plot**

K=2 stress=10.002

**% variance** and Species Scores

- No true % variation explained in this analysis.
- Overall stress is analogous to total percent variation explained.
- Stress for each axis (correlate original distance matrix with pairwise sample similarity on an axis) analogous to variation for an axis.
- Species information is lost when forming the distance matrix.
- Species can be plotted in ordination space by calculating weighted averages.
- See code and notes for PCoA where we did the same for that analysis.
Multiple Starting Points avoid local optima

- Function `metaMDS` (vegan package) uses multiple starting points:
  - Multiple random starting points (default)
  - Performs other ordination first to get a starting configuration

NMDS Code

- `metaMDS` (vegan package)
  - uses multiple starting positions and Procrustes to track convergence
- Code
  ```r
  metaMDS(community, k = 4, distance = "bray")
  ```
- Options
  - `Plot` – plot Procrustes errors along the way
  - `Trymax` – maximum number of starting points
  - `K` – number of axes
  - `Distance` – distance measure to use on raw data
  - `Autotransform` – use some automatic transformations if the analysis thinks they are necessary
  - `Expand` – get species scores as weighted averages
  - `Noshare` – proportion of samples with shared 0 to initiate a correction. Default value is 0.1 (10%) set this to 1.0 to “turn it off”.
  - Various other options to center/rotate scores (metaMDS for details)
- Output
  - Stress values
  - Sample scores in ordination space
  - Species weighted average scores

NMDS Example

Plain isoMDS with species weighted average scores added

No indication of percent variation accounted for on the two axes.
The first axis very nicely captures the pattern in the raw data.

By far the best at capturing the gradient set up in our dataset.

Much improved stress value over isoMDS.

Assignment

- Reading – section 5.6 in text
- Assignment: Community dataset from that paper, found on datadryad [https://doi.org/10.5061/dryad.5qp96](https://doi.org/10.5061/dryad.5qp96)
  - Using the ShortTerm dataset (you will need to save it as csv, may want to add a column for groups), perform the NMDS in figure 5a (using Bray Curtis distance calculated from untransformed data, K=2).
  - Perform ANOSIM to test for differences among dates
  - Plot points by dates (as in figure 5a), and add weighted average scores for all species.
  - Report stress value and estimate % variation on axes 1-2
  - Which sites are contributing the most stress?
  - Repeat all of the above with K=4. What has and has not changed? Explain why.