**PCA (rda function) output**

- prin_comp$rda$community
- summary(prin_comp)

```r
totals $Xbar$
```

- Within the object returned by the function rda(), results are within CA object (`prin_comp$CA`):
  - prin_comp$CA contains:
    - eig: eigenvalues
    - u: site scores
    - v: variable scores (eigenvectors)
    - u.eig: site scores scaled by eigenvalues
    - v.eig: variable scores scaled by eigenvalues
    - Tot.chi: total inertia
    - Xbar: centered and scaled data matrix

```r
prin_comp$CA
```

- Since Xbar contains the centered and scaled data matrix, we could use the eigen() function on a variance/covariance matrix to check the eigenvalues and vectors

```r
eigen(cov(prin_comp$CA$Xbar))
```

- What are “scores”
  - Site scores = Centered and scaled data *%* eigenvectors

**Biplots from different PCA functions, represent different scaling**

- PCA (rda function) output
- Since Xbar contains the centered and scaled data matrix, we could use the eigen() function on a variance/covariance matrix to check the eigenvalues and vectors

```r
eigen(cov(prin_comp$CA$Xbar))
```

- What are “scores”
- Site scores = Centered and scaled data *%* eigenvectors

**Partitioning of variance:**  
**Source Proportion**

<table>
<thead>
<tr>
<th>Source</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>1.000000</td>
</tr>
<tr>
<td>Reduced</td>
<td>0.407264</td>
</tr>
</tbody>
</table>

- Eigenvalues, and their contributions to the variance

<table>
<thead>
<tr>
<th>Eigenvalues</th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
<th>PC5</th>
<th>PC6</th>
<th>PC7</th>
<th>PC8</th>
<th>PC9</th>
<th>PC10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4244062</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2080898</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4244062</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2080898</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1452753</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3843255</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3705668</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2122798</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4087218</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2459313</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Same analysis but with scale=TRUE

**Partitioning of correlations:**  
**Source Proportion**

<table>
<thead>
<tr>
<th>Source</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>1.000000</td>
</tr>
<tr>
<td>Reduced</td>
<td>0.407264</td>
</tr>
</tbody>
</table>

- Eigenvalues, and their contributions to the correlations

<table>
<thead>
<tr>
<th>Eigenvalues</th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
<th>PC5</th>
<th>PC6</th>
<th>PC7</th>
<th>PC8</th>
<th>PC9</th>
<th>PC10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4244062</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2080898</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4244062</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2080898</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1452753</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3843255</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3705668</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2122798</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4087218</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2459313</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

3/20/2016
How many axes to use?

- Function `screeplot()` will plot the variation explained by axis
- Option `bs=TRUE` will include the broken stick distribution of variance explained.
- Recommendation is to ignore axes where variance explained is less than broken stick.

Axes Meanings

- PCA on fish morphological data. Six species of Profundulus, 11 morphological measures (body depth, head depth, dead width, caudal peduncle depth, eye diameter, snout length…)

Species appear different, where are the differences?
What do these axes represent?
How well does the analysis find and summarize structure?

<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-zeroes</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>$\chi^2$</td>
<td>Non-negative, quantitative or binary</td>
</tr>
<tr>
<td>FA</td>
<td>Euclidean</td>
<td>Quantitative, linear relationships assumed, beware of double-zeroes</td>
</tr>
</tbody>
</table>

Axis Rotation

- PCA constructs the new variables (axes) to maximize variance explained. Loadings are used to relate variables to axes.

- What if you want the relationship between variables and axes to be clearer?
  
  - You can have the analysis "rotate" axes to maximize loadings with variables.

  - The tradeoff is that less overall variance will be explained
Factor Analysis

- Function `factanal()`
  - Data matrix
  - Number of factors
  - Method of calculating scores
  - Type of rotation (none = PCA)
  - First, try factor analysis with no rotation. Results look very similar to a PCA

```r
fac <- factanal(community, 6, scores="regression", rotation="none")

fac$scores[, 1]
fac$scores[, 2]
```

- Type of rotation (varimax)

```r
fac <- factanal(community, 6, scores="regression", rotation="varimax")

Loadings:

```
Factor1  Factor2  Factor3  Factor4  Factor5  Factor6
sp1     -0.419   0.153   0.849   0.121   0.244
sp2     -0.178   0.507   0.683   0.441   0.130
sp3      0.210   0.762   0.394   0.460
sp4      0.647   0.687   0.147   0.211
        -0.101
sp5      0.950   0.276
        -0.111
sp6      0.950
        -0.276       0.111
sp7      0.647
        -0.687       0.147   0.211
sp8      0.210
        -0.762       0.394   0.460
sp9     -0.178
        -0.507       0.683   0.441   0.130
sp10    -0.419
        -0.153       0.849   0.121   0.244
```

Non Metric Multidimensional Scaling

- 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

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.

- 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.

- 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

```
> nmds<-isoMDS(distance, k=2, tol=0.001)
initial value 14.250589
iter   5 value 9.263973
iter  10 value 7.920977
iter  15 value 7.734398
final value 7.701990
converged
```
- larger tolerance = convergence reached earlier with fewer iterations.
- Tradeoff in computing time.

- Note that isoMDS uses one starting configuration (susceptible to local optima problem)
Stress relations to # iterations and # axes

Stress Plots

Stress Plot

K=2 stress=10.002

Original Bray Curtis Distance

"% variation" 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.

If you want a good 2D representation, k=2 is better than k=10 even though the stress will be higher.
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

**Stress**

Multidimensional Space

**NMDS Code**

- `metaMDS` (vegan package)
  - uses multiple starting positions and Procrustes to track convergence

**Code**

```
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

**Assignment**

- Reading – section 5.6 in text

- Community dataset (spaeth.csv)
  - Perform NMDS using `metMDS` function
  - Report both types of stress values
  - Estimate % variation overall and on both axes
  - Report goodness of fit for sites (which sites are the worst fit)
  - Plot ordination with species weighted average scores
  - Justify and/or explain treatment of data (transformations etc., similarity index used and number of dimensions used)