**Principal Components Analysis & Redundancy Analysis**

Ecologists have long sought efficient ways to summarize entire datasets, either with respect to specific environmental data, or simply to summarize the distribution of sample points in a low-dimensional space where alternative environmental explanations can be explored. We will do both. First, we will use Principal Components Analysis (PCA) to reduce the distribution of sample points to a few dimensions and plot the results, allowing us to identify patterns that can be modeled linearly. (Put another way, PCA allows us to identify the variables that are responsible for most of the patterns in our dataset.) Depending on how successful we are at reducing the data set, we can then use that information to seek patterns among the distribution of plots in ordination space and explore possible environmental correlates with these via Redundancy Analysis (RDA). (In other words, we will use the PCA axes as dependent variables for RDA to explore how environmental variables affect site x species patterns.) It's important to remember, however, that at most we can establish only correlation among variables, not causation. It is therefore possible that the variables we measured are only surrogates for the real underlying drivers.

PCA takes a “cloud” of data and projects it onto a new, low-dimensional space, which maximizes the variance explained by the first axis. The second axis is projected in a similar manner but in such a way that it is orthogonal to the first axis. (Subsequent axes follow suit, with less variance explained for each axis.) There are as many PCA axes as there are samples in the original data, but usually only the first two are sufficient.

To do this process, PCA creates new vectors that maximize their linear correlation with all the other variables.

PCA is a form of unconstrained ordination (indirect gradient analysis) whereas RDA is a form of constrained ordination (direct gradient analysis). If you have both site x species and site x environmental data, using both in tandem is possible, as we will do.

Open a new RStudio session (with your class working directory) with the following libraries:

*labdsv*

*MASS*

*MVA*

*optpart*

*picante*

*stats*

*vegan*

Load the bryceveg.R data as veg and brycesite.R as site (both with header = TRUE).

**PCA: to reduce dimensionality of a large dataset of interrelated variables**

PCA is perhaps the most commonly used form of ordination, and is used in a variety of disciplines. The aim of PCA is to calculate new, synthetic variables called principal components, created via matrix operations on the original dataset. Each principal component is a linear combination of the original variables that are oriented in directions that describe maximum variation among individual sampling entities (i.e., it organizes entities along continuous gradients defined by the principal components and describes the sources of greater variation among entities). The first principal component represents an axis in multidimensional data space that produces the largest dispersion of values. Subsequent principal components are calculated as orthogonal to each previous one and account for the remaining scatter of the data values. The first axis (component) thus represents the largest gradient of variability in the dataset, the second axis represents the second largest, and so on until all of the variability in the data has been accounted for. The strongest covariation among variables emerges in the first few (usually 1-3) axes (components), hence the name “principal components analysis.” These operations are performed on your data

Looking at the % of the total variance that is explained by each axis can reveal whether there are any dominant variables or gradients in the dataset. We can also infer whether few (first few axes explain most of the variance) or many (each axis makes a near-equal or only slightly less contribution) variables are influential. The result is a reduced dataset from *n* objects and *p* variables to a smaller number of synthetic variables that represent most of the information in the original dataset.

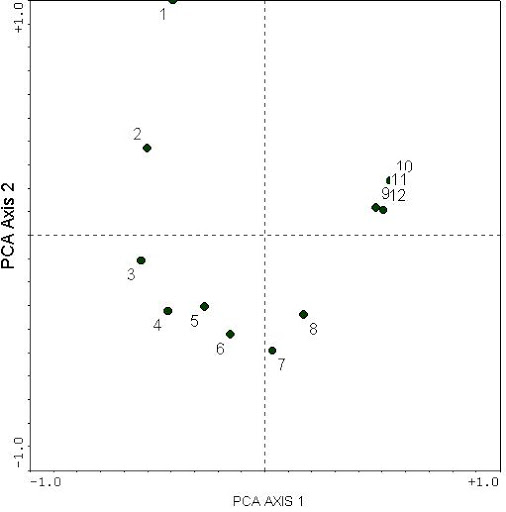
There is a related technique called Principal Coordinates Analysis (PCO or PCoA) that is similar to PCA: whereas PCA organizes objects by eigenanalysis of a correlation or covariance matrix of Euclidean distances, PCoA can be applied to any distance/dissimilarity matrix. We do not have time to cover PCoA here, but it is very useful in dealing with phylogenetic distances. See the pcoa() function in package *ape*.

**Assumptions:**

Developed by Karl Pearson in 1901, PCA is most effective for data with approximately **linear** relationships among variables. Because linear relationships are relatively, rare, however, most examples of PCA in community ecology exemplify inappropriate use!

Because of this limiting assumption, **PCA is best-suited for VERY short environmental gradients and thus is probably best-suited to studying within-community variation rather than variation among communities**. (With longer gradients, other ordination methods [especially Detrended Correspondence Analysis or Correspondence Analysis/Reciprocal Averaging] are recommended. It may still be possible to use PCA by linearizing your data via a log transformation or use of a power model. See Tukey [1977] for his “ladder of transformations” [suggestions of transformations to try to linearize data]. Multivariate normality is not strictly needed if your goals are descriptive.) In the coming weeks, we will cover an alternative to PCA for examining species x site x environment relationships out of a very long list of ordinations; one to recommend is Correspondence Analysis). PCA is primarily used now as a way of detecting independent combinations of influential variables.

PCA **does not perform well if your dataset has many 0’s**. If PCA is done on datasets with many 0’s or over long environmental gradients, it can result in what’s called a **horseshoe effect** (whereby objects at the edges of the environmental gradient appear close to each other in ordination space even though they’re dissimilar). (If you had used a Chord or Hellenger transformation on your data to deal with lots of 0’s, this effect is minimized.)



Example of the horseshoe effect from Mike Palmer, Oklahoma State University:

In the case of such datasets, then Correspondence Analysis (CA) is recommended over PCA. (We will not have time to cover Correspondence Analysis in this course, but here are a few key points: PCA maximizes the amount of explain variance among variables whereas CA maximizes the correspondence (similarity of frequencies) between rows (variables) and columns (objects). PCA assumes a linear relationship among variables whereas CA assumes a unimodal distribution. CA is especially useful for datasets of species composition and abundances, which typically have lots of 0s. It does not suffer from the horseshoe effect but has a problem of its own called an arch effect [which, if present, can mean you need to use Detrended Correspondence Analysis]. For more info, see the ca() function in package *MVA*.) (We will be covering Canonical Correspondence Analysis later, but that is not the same thing!) (Both the arch and horseshoe look similar, but the horseshoe typically has curved-in ends.)

So examine your data for sparsity, check for linearity, and eliminate outliers; transform the data as needed first.

PCA is used when there are numerous, correlated variables to weed out redundant variables and generate new, fully uncorrelated variables (PC axes) that represent combinations of original variables. The resulting PCs may then be used in statistical tests like ANOVA in lieu of the original, correlated variables.

To do PCA, data are usually in the following form (samples-by-variables matrix):

P1 P2 … PP

1 x11 x12 x1P where there are N samples and

2 x21 x22 x2P P variables (N can be sites, habitats,

… or niches; P can be environmental

N xN1 xN2 xNP characteristics, abundances, or behaviors).

Those PCs with the largest eigenvalues contribute the most variation, and the variables with the largest eigenvectors or **component loadings** (a.k.a. **factor loadings**) (positive or negative) contribute the most to that PC. Example:

PC1 PC2 PC3 PC4

Eigenvalue 25.3095 3.4170 1.0001 0.6742

The first PC explains the most variation in the data. PCs 2-4 are relatively irrelevant. (Important: the eigenvalues are NOT the % variance explained; they are used to calculate % variance explained.) Most variation in this example is left unexplained, meaning that perhaps additional variables need to be measured. Eigenvectors represent the gradients of data dispersion in ordination space and are used as the ordination gradients (axes); the eigenvalues designate the strength of each gradient.

Then examine the eigenvectors:

Eigenvector PC1 PC2 PC3 PC4

Forb -0.6421 0.1124 -0.0012 0.0014

Grass 0.4111 0.0991 0.0001 0.0122

Fern 0.0013 -0.0009 0.5587 0.1224

Pine 0.1499 0.4578 0.1200 0.0700

Oak 0.0009 -0.0110 0.3887 0.0499

PC1 appears to reflect ground cover (lack of forbs and presence of grasses), PC2 reflects pine forests, PC3 represents mesic environments (ferns and oaks), and PC4 is pitiful. Recall that only PC1, however, was the most influential. Therefore, variables other than forb and grass may be eliminated. It appears that presence/absence of forbs and grasses is important.

Typically, of greatest interest are the PCA outputs of:

* variance explained
* cumulative variance explained
* species loadings
* site scores
* plots (especially biplots)

**Example:**

(For a step-by-step example involving the math, see McCune and Grace 2002.)

PCA is an **eigenanalysis** technique that attempts to explain as much variance as possible on each of a series of orthogonal vectors spanning the data space. Eigenanalysis is based in linear or matrix algebra, and has a wide range of uses. It requires a square, symmetric data matrix. (A square matrix has the same number of rows as columns. A symmetric matrix is a type of square matrix that is the same if you transpose rows and columns.) You can get exact results only for very small matrices (typically < 3 rows and columns). For large matrices, eigenanalysis requires an iterative approach that eventually "closes in" on the answer. The results of an eigenanalysis is a series of **eigenvalues** and **eigenvectors**. Each eigenvalue has an eigenvector, and there are as many eigenvectors and eigenvalues as there are rows in the initial matrix. Mathematically, eigenvectors are the directions along which a linear transformation acts; eigenvalues refer to the magnitude of the transformation in the direction of eigenvector. In ecology, an eigenvector represents the gradients of data and are used as ordination axes. An eigenvalue is a measure of the strength of that gradient (ordination axis) based on the amount of variation along the axis and thus assesses the importance of an ecological gradient; it represents the variance in the community matrix that is attributed to a particular axis.

Our particular interest is in finding an alternative description of our data in a low-dimensional space. The first vector will account for as much variance in plot dispersion from the centroid as possible; the second will be chosen by the same criterion but subject to the constraint of being orthogonal the first, and so on. The approach can be likened to regression by least squares, except that the residuals from the vector are measured perpendicular to the vector rather than perpendicular to the X or Y axis.

In general, PCA is performed on a correlation or covariance matrix, although equivalently, a matrix of sums-of-squares and-cross-products can be used. In R, however, the functions we will use convert the basic vegetation data into the appropriate forms, usually requiring only an argument to be specified in the function.

There are several ways to perform PCA in R: princomp() and prcomp() in stats are the most common ways. princomp() computes an eigenanalysis and prcomp() computes a singular value decomposition. You cannot use princomp() if you have fewer sites than species, as in our case, so we cannot use that function here. You can also use pca() from the *labdsv* package. Each of these three methods includes different plotting options.

PCA accepts site x species data matrices or dataframes and computes **loadings** (also called weights) for each column and **scores** for each row. The loadings are the contribution of the column vector to each of the eigenvectors. A large positive component means that that column (species in our case) is positively correlated with that eigenvector; a large negative values is negative correlation; and small values mean that the species is unrelated to that eigenvector. Scores for the sampling units (sites) are calculated by multiplying the weights by the original abundance values for each response and then summing the result for each site. The variation reflected by each axis of site scores is the proportion of the variation in the original response matrix (% of variance explained, see below).

PCA allows you to specify a number of parameters concerning the calculation, the first being whether you want to use a correlation or covariance matrix.

A PCA without dividing by the standard deviation is an eigenanalysis of the covariance matrix, and a PCA in which you do indeed divide by the standard deviation is an eigenanalysis of the correlation matrix. **When using species/variables measured in different units, you should use a correlation matrix (because doing so is equivalent to standardizing each of the variables to mean = 0 and standard deviation = 1); when the variable scales are similar you can use covariance.** This is a subject too detailed to expound on adequately in this lesson, but generally speaking, PCA is sensitive to the scale of measurement of your data. If all the data are not measured on the same scale, using covariance means that the result will be determined mostly by the variable with the largest values, as it will have the highest variance. Using a correlation matrix treats all variables the same (standardized to mean = 0 and std. dev. = 1). Even if all species were measured on the same scale (e.g. percent cover), **to prevent the dominant species from determining the results, you probably want to use correlation**. In pca(), this means specifying cor=TRUE in the function call. For prcomp(), you use scale=TRUE (scale = FALSE [default] is covariance).

By default, pca() will generate one eigenvector for every column (species), and scores for each row (plot) on each of the eigenvectors. Generally, the vast majority of the variance is described on the first few eigenvectors, and we can save space by only calculating the scores for only the first few eigenvectors. This can be specified by including dim=n where n equals the number of dimensions you want scores and loadings for. So, for example

pca.1 <- pca(veg, cor=TRUE, dim=10)

calculates a principal components analysis of the Bryce Canyon vegetation, using a correlation matrix and only calculating scores for the first 10 eigenvectors.

So now let us do a PCA on the full Bryce Canyon vegetation dataset, using prcomp():

pca.2 <- prcomp(veg, scale = FALSE)

The scale = FALSE argument (default) does not scale the variables to comparable variance. Having scale = TRUE is very useful if you have columns of data that are measured at different scales. (In this case, the species have been measured at the same quadrat scale.)

**Variance explained:**

To see the variance and cumulative variances explained by eigenvector, type:

summary(pca.2)

The output looks like this:

Importance of components:

PC1 PC2 PC3 PC4 PC5 PC6

Standard deviation 1.7995 1.0904 0.79282 0.64676 0.55666 0.52348

Proportion of Variance 0.3504 0.1286 0.06801 0.04526 0.03353 0.02965

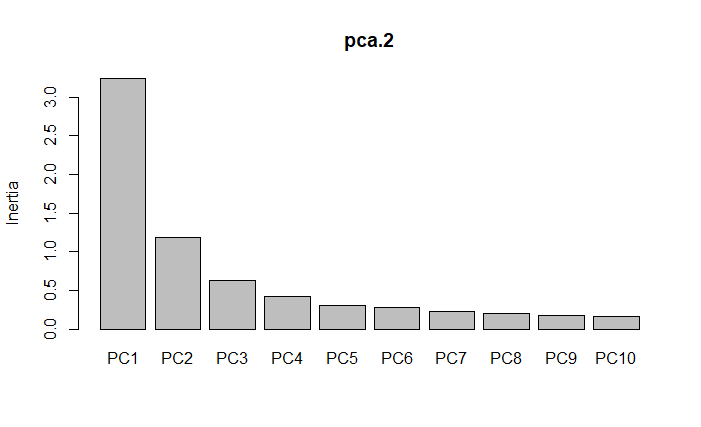
Cumulative Proportion 0.3504 0.4790 0.54703 0.59229 0.62582 0.65547

Etc.

For reasons known only to R, the top line is the standard deviation (rather than the variance) associated with each component (i.e., each axis). The next two lines are the items of more interest: proportion of total variance and cumulative proportion of variance. Notice that the variance explained is in order from highest to lowest by eigenvector; this is by design. Notice also that the first two eigenvectors explain ~48% of the total variance of the dataset. In general, if your first two PC axes don’t explain at least 60% of the variance, then PCA is not an informative analysis because there is just too much unexplained variance due to factors not measured in your study (Hair et al. 2018). So this isn’t a great example, but let’s run with it for illustrative purposes.

In every PCA, PC1 explains the most variance, then PC2, and so on. You can visualize this as a **scree plot**:

screeplot(pca.2)



If you want to see all the PC axes:

barplot(summary(pca.2)$importance[2,], las=2)

title(ylab="Proportion of variance", xlab="PCA axis")

But in this example, the amount of variation being explained by the first few axes is really low.

So now we need to determine what variables (sites and species) are associated most with PC1 and PC2.

**Species loadings:**

To see the species loadings (eigenvectors), prcomp() uses the term rotation:

loadings <- pca.2$rotation

loadings

A partial output looks like the following:

PC1 PC2 PC3 PC4

junost -0.0063846863 0.0008985748 0.0259060952 4.630351e-03

ameuta 0.0398925055 0.0040519709 -0.0188452937 3.968258e-02

arcpat 0.8551732366 -0.3986456291 0.2464910430 -2.602104e-02

arttri -0.0580591176 0.1972214853 0.5877646517 6.331815e-02

atrcan -0.0378689915 0.1292042139 0.3483085167 3.803780e-02

berfre -0.0010397366 0.0121286157 0.0257395588 -7.459811e-03

Etc.

You can see that eigenvector 1 is negatively correlated with junost, arttri, atrcan, and berfre, and positively associated with ameuta and arcpat (along with many more species not included in this excerpt). All of these values are small, meaning that these species are mostly unrelated to PC1. If we only had these species, PC1 would be most influenced by abundance of arcpat and PC2 by its absence.

(Note: it is possible that loadings or scores from two different computers or runs of the same data will come out with the same values but with opposite signs. The orientation of eigenvectors is arbitrary, and the sign is only meaningful with respect to other values on the same PC axis.)

Because we have 169 species, I won’t go through all of them, but what you ought to do is examine what species are most strongly associated (positively and negatively) with your most influential PC axes. If we had a dataset that was just the 6 species above from 4 sites, then we could conclude that PC1 is an indicator of the presence of arcpat whereas PC2 is associated with lower abundances of that species.

**Site scores:**

For site scores:

scores <- pca.2$x

scores

which looks like:

PC1 PC2 PC3 PC4 PC5

bcnp\_\_1 0.303522539 0.58234332 -0.4964863489 0.446266093 -0.0005421279

bcnp\_\_2 -0.106991234 0.86395259 -0.7710460128 1.070231898 0.1001168298

bcnp\_\_3 0.325220583 0.63188537 -0.5977936134 0.816265090 0.0339421175

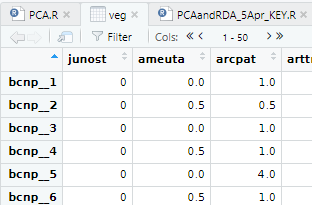
bcnp\_\_4 0.301010236 0.55085413 -0.4413011878 0.356285781 -0.0722309177

bcnp\_\_5 3.051327186 -0.58628755 0.0316738283 0.669026942 -0.0280048460

Etc.

As before with species, we need to determine which sites are most associated with our most important PC axes. Since we had 160 sites, I won’t go through all of them; instead, assume we just had the 5 sites above. PC1 is most associated with site bcnp\_5; PC2 is most associated with site bcnp\_2.

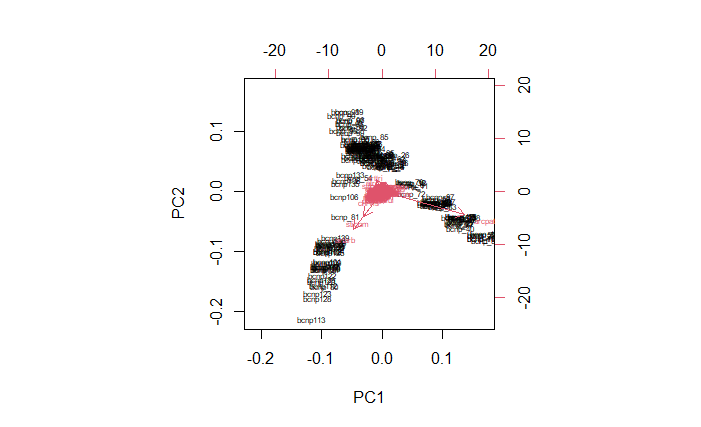
So given the species results we had before (where PC1 => arcpat higher abundance, PC2 => arcpat lower abundance), we can conclude that arcpat should be present in higher numbers at site bcnp\_5 than at site bcnp\_2…and if you examine veg, you can see that that is indeed the case:



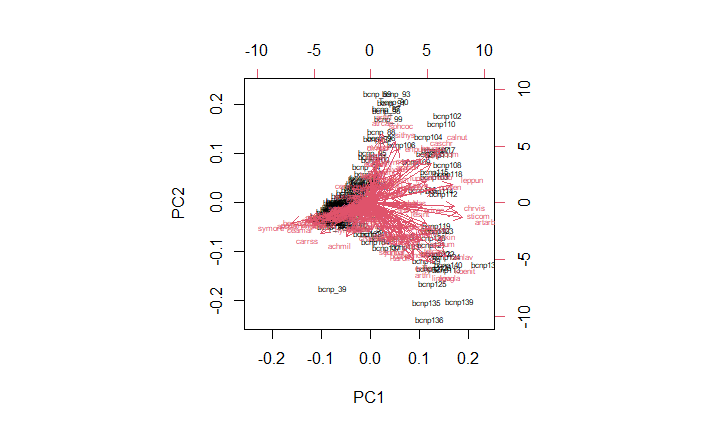
This example of 6 species and 5 sites is trivial but is an example of how you need to examine and interpret your output.

The scores are what is typically plotted in a PCA ordination. If you used prcomp(), then

biplot(pca.2, cex=0.5)



The sites are in black text, species in pink with arrows. Longer arrows indicate higher abundance. This graph shows a very strong **horseshoe effect**, likely due to the large number of 0s in the dataset. If you redo all of the above analyses for pca.2 with scale = TRUE, you get this:



The loadings and scores will be different, of course and, thus, so will be your conclusions.

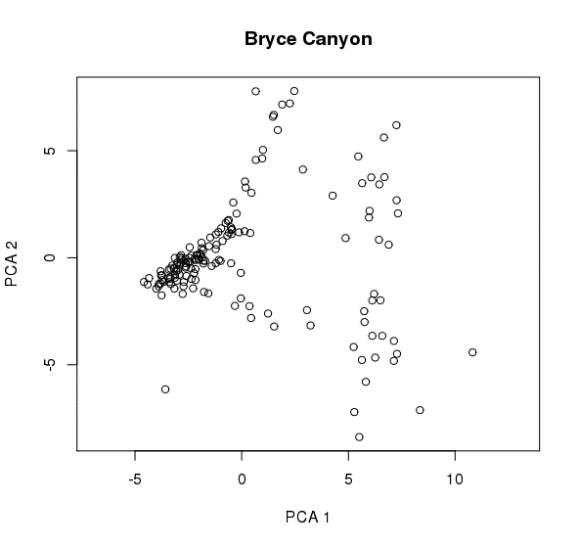
In this biplot, you can see how certain species are strongly associated with the conditions found at certain sites. Notice how most of the species’ arrows point right, meaning they are positively associated with PC1 (a few species negatively associated). There are about as many positive as negative associations with PC2.

If you used pca()from *labdsv*, then:

plot(pca.1,title="Bryce Canyon")



yields a graph with circles as sites in the same orientation as the black text in our previous biplot, but it lacks the species info (because a plot isn’t a biplot!). Without labels, this graph is not very helpful, but it shows how there is a group of sites that are rather similar in community composition. Moreover, see how there is a distinction in sites along PC1 (blue circles below):



There is no such pattern of distinction that can be discerned along PC2. This is not surprising, given that PC2 explains less variation than does PC1, and given how little variation they explained, I’m surprised there’s any pattern that can be seen at all!

Because prcomp() has a biplot option, it is more commonly used than pca().

Now that we have some useful axes of variation (the principal components), we can use them in other analyses about the influence of environment on species. That means we need a constrained form of ordination.

**Redundancy Analysis: exploring relationships with environmental variables**

Of course the whole point of getting the first few axes of variation is to begin an analysis of biotic community/environment relationships. There are several ways of doing so, one of which is called Redundancy Analysis (described in detail by Legendre and Legendre 1998). Unlike PCA, it’s a constrained ordination that is used to determine how much of the variation in one set of variables can be explained by the variation in another set of variables.

Whereas PCA calculates the synthetic axes that minimize the total error sum of squares for a linear combination of the response variables, RDA calculates the synthetic axes that have the best linear combination of the response variables with the provided explanatory variables.

**Assumptions:**

Like PCA, RDA assumes linear relationships among variables (indeed, RDA is a canonical version of PCA where the principal components are linear combinations of the explanatory variables). Thus, RDA is subject to the same limitations as mentioned earlier for PCA. (If your variables exhibit a unimodal rather than linear relationship, then do a Correspondence Analysis (unconstrained) to start and then follow up with a Canonical Correspondence Analysis (constrained), which we’ll cover in an upcoming lesson.)

For any type of constrained ordination, you need two datasets: 1) a dataset with the response (dependent) variables (e.g. species presence or abundance); 2) a dataset with the explanatory (predictive) variables (e.g. environmental variables measured at the same sites as for dataset 1).

**Example:**

There are multiple ways of doing RDA in R. We will use rda() in package *vegan* because it is commonly used.

First, just as we did with PCA, we will want to scale the data. To center the species abundances around their means and standardize their variance you can use:

veg.center <- apply(veg,2,function(x){(x-mean(x))/sd(x)})

The apply function applies a function to each column (if we specify 2 as the second argument, or to rows if we specify 1) in turn. The function(x){x-mean(x)} says to subtract the mean value of each column from every value in that column, and then divide by the standard deviation of that column. If you don't divide by the standard deviation, you will get results equivalent to calculating the PCA on a covariance matrix rather than a correlation matrix. (See arguments for why you’d want to use one or the other on pp. 5-6.)

Alternatively, we can use a function in *labdsv* called scale():

veg.center <- scale(veg,center=TRUE,scale=TRUE)

(This yields the same result as the previous line of code but handles missing values better.)

The center=TRUE argument means to center the columns by their mean, and the scale=TRUE means divide by the standard deviation.

(Earlier, in the PCA section of this lesson, we used scale=TRUE to scale the data but did not specify anything about center; that was because center=TRUE is the default. I’m including it here just to explain it, but it’s redundant. If you do include it, put it before scale.)

To specify which (numeric) environmental variables from site that we want to use, and to center them:

attach(site)

(The attach() command allows you to call variables from site without naming site every time.)

env <- cbind(elev,av,slope)

env.scale <- scale(env,center=TRUE,scale=TRUE)

This creates a new data matrix with elevation, aspect value, and slope in in it, which we then center and scale to unit standard deviation. We did so because the environmental data represent the independent variables and were measured on different scales; we want to both center and scale to unit standard deviation to simplify interpretation.

Now that our site x species and site x environment data are scaled and centered, let’s take rda() in *vegan* for a spin:

RDAdemo <- rda(veg.center)

RDAdemo

Call: rda(X = veg.center)

Inertia Rank

Total 169

Unconstrained 169 148

Inertia is variance

Eigenvalues for unconstrained axes:

PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8

13.343 7.685 7.076 6.482 5.694 5.357 4.486 4.273

(Showing 8 of 148 unconstrained eigenvalues)

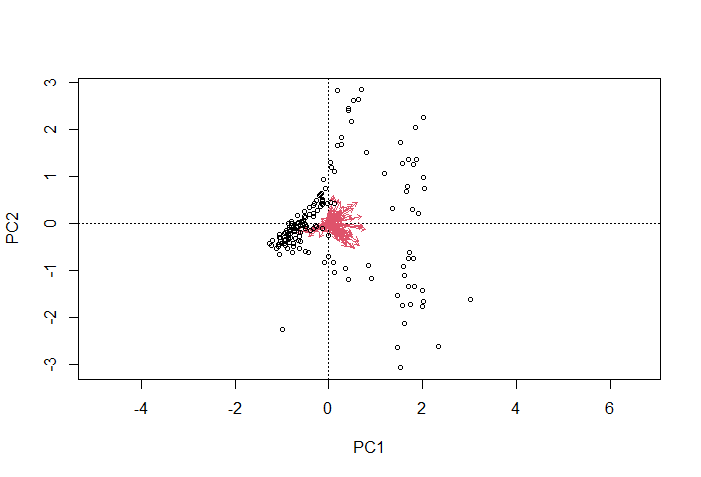
summary(RDAdemo)

The eigenvalues show the proportion of variance explained; from the summary() statement, you can see that the first two PC axes explain ~12% of the overall variance in the data. This is low; ideally, you’d like to see at least 50% explained in the first two axes. But given that there are so many PC axes in our data (because we have so many sites), this is perhaps to be expected. To visualize this:

screeplot(RDAdemo)

You can see how there is no sharp drop-off in the variance explained among the first few axes. If you then make a biplot:

biplot(RDAdemo)

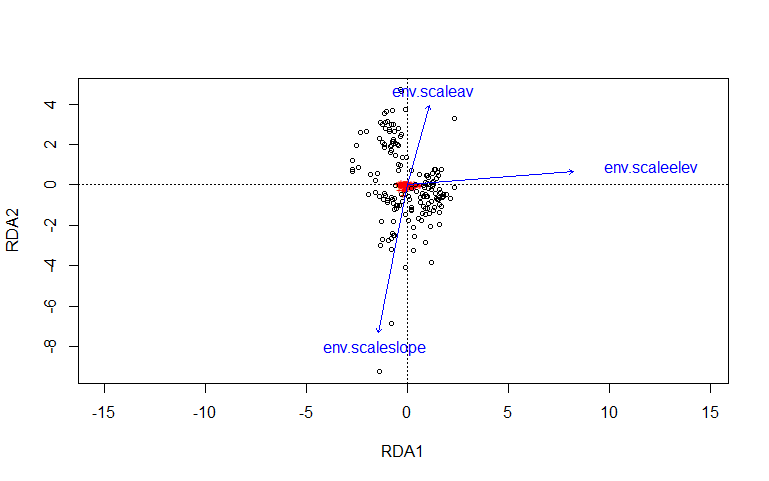


the result is a PCA (essentially the same as the biplot generated by PCA with prcomp() on p. 10), because remember that RDA is PCA (a constrained PCA). In practice, I find rda()’s biplots to be less informative than prcomp()’s and so recommend that you use a dedicated PCA function (like prcomp()) to make a biplot rather than rda().

So now we can use these ordinated axes in an RDA that examines how elevation, aspect value, and slope structure vegetation in Bryce Canyon:

RDAdemo2 <- rda(veg.center ~ env.scale, site)

plot(RDAdemo2)



In this **triplot**, you cannot make out any patterns with respect to species (the red clot in the center of the plot), as their variation is swamped by variation in the sites that “stretches” the plot. The sites display far more variation along both PC axes, but with greater separation of sites along RDA2. Arrows that are parallel to an axis are more strongly associated with that axis, and arrow length indicates correlation strength. You can see how aspect and slope are negatively associated with each other along RDA2, and how RDA1 is associated with increasing elevation. Thus, sites in Bryce Canyon differ more in slope and aspect than they do in elevation. The plants, however, are not strongly associated with these variables.

When we take these results and consider how little variation was explained in the PCA axes, it’s clear that factors other than elevation, aspect, and slope are at play.

**Summary:** PCA and RDA are very similar (are related analyses). PCA is an unconstrained form of ordination (it searches for any variable(s) that best explains species composition), whereas RDA is constrained (it searches for the best explanatory variables). PCA can help winnow down a large number of correlated variables to a smaller set, which can then be used in a subsequent RDA (or other constrained ordination). Performed sequentially, they are a powerful pairing in detecting species-environment relationships.

**References:**

Hair, J.F., B.J. Babin, R.E. Anderson, and W.C. Black. 2018. *Multivariate Data Analysis (8th Ed.)*. Cengage, Boston, MA.

Legendre, P., and L. Legendre. 1998. *Numerical Ecology*, 2nd ed. Elsevier, New York, NY.

McCune, B., and J.B. Grace. 2002. *Analysis of Ecological Communities*. MJM, Gleneden Beach, OR.

Tukey, J.W. 1977. *Exploratory Data Analysis*. Pearson, New York, NY.

**Assignment:** due 0800 Monday, April 5

Start a fresh RStudio session. Remember to set your working directory to your course folder and use the same package libraries as we used today.

Working with Ground\_beetles\_abundance.csv (row.names=1) and GBsite.csv (header=TRUE), use what you learned today to answer the following questions:

**Q1. Do the first two principal components represent at least 60% of the variance in the beetle abundance dataset? (Be sure to decide whether the data should be scaled/centered first.)**

**Q2. For the first two axes, what sites contribute the most to the overall variance of the beetle abundance dataset, and in what manner (positive/negative)? Given that the site names include info about which of three habitat types they represent (Grass, Wood, Edge), are there any patterns you can discern in this?**

**Q3. For the first two axes, what species contribute the most to the overall variance of the beetle abundance dataset, and in what manner (+/-)?**

**Q4. Generate and interpret a biplot of the beetle site x species data.**

**Q5. Now perform a constrained analysis on the beetle data to examine the effect of maximum vegetation height in structuring beetles at our 18 sites/3 habitat types.**

Make an RMarkdown Word file of your work and turn that in. Be sure to include your answers to any questions asked! Turn in your assignment as a Word document via email to [iroro.tanshi@ttu.edu](mailto:iroro.tanshi@ttu.edu) no later than 8:00 a.m. on Monday of next week. In your email, please include the following as the Subject line:

Assignment on PCA/RDA