PRINCIPAL COMPONENTS ANALYSIS (PCA)

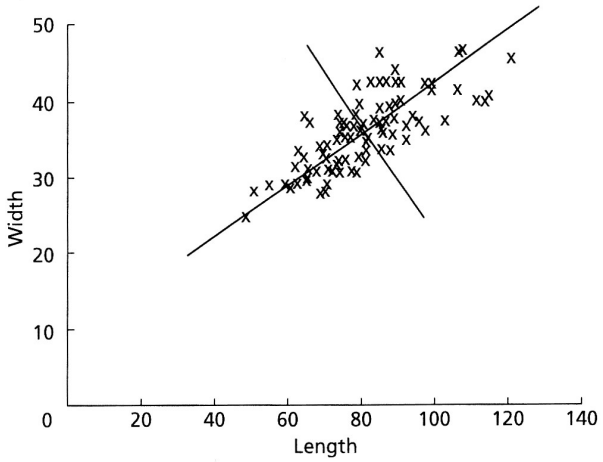
Steven M. Holland

Department of Geology, University of Georgia, Athens, GA 30602-2501

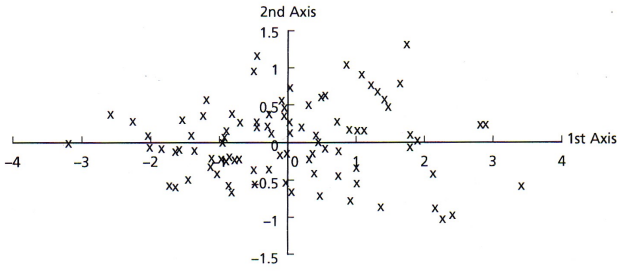
5 December 2019

Introduction

Suppose we had measured two variables, length and width, and plotted them as shown below. Both variables have approximately the same variance and they are highly correlated with one another. We could pass a vector through the long axis of the cloud of points and a second vector at right angles to the first, with both vectors passing through the centroid of the data.



Once we have made these vectors, we could find the coordinates of all of the data points relative to these two perpendicular vectors and re-plot the data, as shown here (both of these figures are from Swan and Sandilands, 1995).



In this new reference frame, note that variance is greater along axis 1 than it is on axis 2. Also note that the spatial relationships of the points are unchanged; this process has merely rotated the data. Finally, note that our new vectors, or axes, are uncorrelated. By performing such a rotation, the new axes might have particular explanations. In this case, axis 1 could be regarded as a size measure, with samples on the left having both small length and width and samples on the right having large length and width. Axis 2 could be regarded as a measure of shape, with samples at any axis 1 position (that is, of a given size) having different length to width ratios. PC axes will generally not coincide exactly with any of the original variables.

Although these relationships may seem obvious, when one is dealing with many variables, this process allows one to assess much more quickly any relationships among variables. For data sets with many variables, the variance of some axes may be great, whereas others may be small, such that they can be ignored. This is known as reducing the dimensionality of a data

set, such that one might start with thirty original variables, but might end with only two or three meaningful axes. The formal name for this approach of rotating data such that each successive axis displays a decreasing among of variance is known as Principal Components Analysis, or PCA. PCA produces linear combinations of the original variables to generate the axes, also known as principal components, or PCs.

Computation

Given a data matrix with p variables and n samples, the data are first centered on the means of each variable. This will insure that the cloud of data is centered on the origin of our principal components, but does not affect the spatial relationships of the data nor the variances along our variables. The first principal components (Y_1) is given by the linear combination of the variables X_1, X_2, ...,X_p,

$$Y_1 = a_{11}X_1 + a_{12}X_2 + \dots a_{1p}X_p$$

or, in matrix notation,

$$Y_1 = a_1^T X$$

The first principal component is calculated such that it accounts for the greatest possible variance in the data set. Of course, one could make the variance of Y_1 as large as possible by choosing large values for the weights a_{11}, a_{12}, ... a_{1p}. To prevent this, weights are calculated with

$$a_{11}^2 + a_{12}^2 + \dots + a_{1p}^2 = 1$$

the constraint that their sum of squares is 1.

The second principal component is calculated in the same way, with the condition that it is uncorrelated with (i.e., perpendicular to) the first principal component and that it accounts for the next highest variance.

$$Y_2 = a_{21}X_1 + a_{22}X_2 + \dots a_{2p}X_p$$

This continues until a total of p principal components have been calculated, equal to the original number of variables. At this point, the sum of the variances of all of the principal components will equal the sum of the variances of all of the variables, that is, all of the original information has been explained or accounted for. Collectively, all of these transformations of the original variables to the principal components is

$$Y = XA$$

Calculating these transformations or weights requires a computer for all all but the smallest matrices. The rows of matrix A are called the eigenvectors of matrix S_x, the variance-covariance matrix of the original data. The elements of an eigenvector are the weights a_{ij}, and are

also known as loadings. The elements in the diagonal of matrix S_y, the variance-covariance matrix of the principal components, are known as the eigenvalues. Eigenvalues are the variance explained by each principal component, and to repeat, are constrained to decrease monotonically from the first principal component to the last. These eigenvalues are commonly plotted on a scree plot to show the decreasing rate at which variance is explained by additional principal components.

The positions of each observation in this new coordinate system of principal components are called scores and are calculated as linear combinations of the original variables and the weights a_{ij}. For example, the score for the r^{th} sample on the k^{th} principal component is calculated as

$$Y_{rk} = a_{1k}x_{r1} + a_{2k}x_{r2} + \cdots + a_{pk}x_{rp}$$

In interpreting the principal components, it is often useful to know the correlations of the original variables with the principal components. The correlation of variable X_i and principal component Y_j is

$$r_{ij} = \sqrt{a_{ij}^2 Var(Y_j)/s_{ii}}$$

Because reduction of dimensionality, that is, focussing on a few principal components versus many variables, is a goal of principal components analysis, several criteria have been proposed for determining how many PCs should be investigated and how many should be ignored. One common criteria is to ignore principal components at the point at which the next PC offers little increase in the total variance explained. A second criteria is to include all those PCs up to a predetermined total percent variance explained, such as 90%. A third standard is to ignore components whose variance explained is less than 1 when a correlation matrix is used or less than the average variance explained when a covariance matrix is used, with the idea being that such a PC offers less than one variable's worth of information. A fourth standard is to ignore the last PCs whose variance explained is all roughly equal.

Principal components are equivalent to major axis regressions. As such, principal components analysis is subject to the same restrictions as regression, in particular multivariate normality. The distributions of each variable should be checked for normality and transforms used where necessary to correct high degrees of skewness in particular. Outliers should be removed from the data set as they can dominate the results of a principal components analysis.

PCA in R

1) For this example, we will use the Nashville carbonates geochemistry data set (Theiling et al. 2007), available on the GEOL 8370 website (http://strata.uga.edu/8370/data/). This data set has geochemical analyses of 200 limestone samples, including the major elements Al, Ca, Fe, Mg, Mn, and Si, stable isotope ratios of carbon and oxygen (d13C and d18O). It also records the stratigraphic position of the samples. As we are most interested in how the geochemical

composition varies, we will pull those variables (columns 2–9) off into a separate data frame for our analysis.

```
> nashville <- read.table(file='NashvilleCarbonates.csv',
header=TRUE, row.names=1, sep=',')
> geochem <- nashville[ , 2:9]
```

Many of the major elements (columns 3–8) are right-skewed, as might be expected for values that cannot go below zero, so a log-transformation is needed.

```
> geochem[ , 3:8] <- log10(geochem[ , 3:8])
```

In general, data sets will need some cleaning before a principal components analysis to analyze only those variables that should be included, to perform any necessary data transformations. Outliers and strongly skewed variables can distort a principal components analysis.

2) Of the several ways to perform an R-mode PCA in R, we will use the **`prcomp()`** function that comes pre-installed in the MASS package. To do a Q-mode PCA, the data set should be transposed first. R-mode PCA examines the correlations or covariances among variables, whereas Q-mode focusses on the correlations or covariances among samples.

```
> pca <- prcomp(geochem, scale.=TRUE)
```

By default, **`prcomp()`** will center the data, that is, move the cloud of data so that its centroid is at the origin. However, **`prcomp()`** will by default perform a principal components analysis on the variance-covariance matrix, which means that the absolute size of each variable will be considered. This will cause variables with larger values to contribute more than variables with smaller values. Although this is sometimes desirable, for this geochemical data we would like all of the variables to have an equal weight; in other words, we want the principal components analysis to be performed on a correlation matrix. To do this, we set the argument **`scale.=TRUE`** (don't forget the period after **`scale`**).

The **`pca`** object holds everything we need (scores, loadings, variances), but the names where these are stored aren't obvious. To keep things clear, it can be helpful to pull these components out into more clearly named objects.

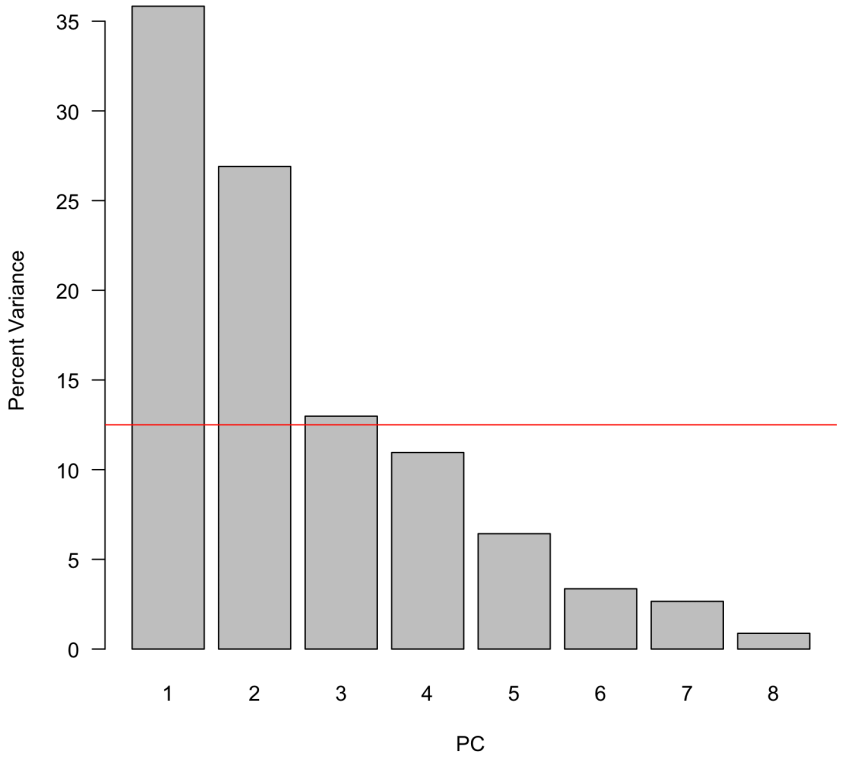
```
> variance <- (pca$sdev)^2
> loadings <- pca$rotation
> rownames(loadings) <- colnames(geochem)
> scores <- pca$x
```

3) A scree plot is useful for understanding how variance is distributed among the principal components, and it should be the first step in analyzing a PCA. The scree plot is particularly critical for determining how many principal components should be interpreted. Although this could be done by calling **`plot(pca)`**, a better-annotated plot that plots percent of total variance for each principal component can be made as follows.

```
> varPercent <- variance/sum(variance) * 100
> barplot(varPercent, xlab='PC', ylab='Percent Variance',
names.arg=1:length(varPercent), las=1, col='gray')
```

One guideline for the number of principal components to use is to accept all principal components that explain more than one variable's worth of data. If all the variables contributed the same variance, this cutoff would be 1/p, where p is the number of variables.

```
> abline(h=1/ncol(geochem)*100, col='red')
```



For this data set, two or possibly three principal components should be examined closely. The first two PCs account for almost 60% of the variance, and there is a large drop-off in percent variance from the second to the third PC. The third PC accounts for slightly more than one variable's worth of information (indicated by the red line), so it may also be worth exploring.

4) A table of loadings should b examined next, as it shows which variables have high loadings (positive or negative) on each principal component, that is, which variables contribute most strongly to each PC. Examining this table can give you a good sense of what each principal component represents, in terms of the original data. A positive loading means that a variable correlates positively with the principal component; a negative loading indicates a negative correlation. Rounding the loadings to two or three decimal places often makes these tables easier to parse.

```
> round(loadings, 2)
```

Since we will be considering only the first two or three PCs, we will display only those.

```
> round(loadings, 2)[ , 1:3]

       PC1   PC2   PC3
d13C -0.17 -0.23  0.64
d18O  0.05  0.60  0.06
Al   -0.50  0.29  0.03
Ca    0.33  0.04  0.47
Fe   -0.53 -0.09 -0.10
Mg   -0.06  0.55  0.44
Mn   -0.23 -0.42  0.39
Si   -0.51  0.09 -0.06
```

Note that Fe, Si, and Al have strong negative loading on axis 1, indicating that higher values on PC 1 correspond to lower concentrations of these elements. On PC 2, d18O and Mg have strong positive loadings, whereas Mn has a strong negative loading.

5) A biplot is a standard way of showing the sample scores and variable loadings in a single plot. Doing this will not only show which samples are similar to one another, but how the variables control this similarity. A **distance biplot** (see Legendre & Legendre, 1998, p. 403) preserves the true distances between samples, but at the cost of distorting (somewhat) the correlations among the variables. A distance biplot can be built with the built-in **`biplot()`** function.

```
> biplot(scores[, 1:2], loadings[, 1:2], cex=0.7)
```

Often, it is more helpful to build this plot manually, which offers more control over its appearance.

```
> plot(scores[, 1], scores[, 2], xlab='PC 1', ylab='PC 2',
type='n', xlim=c(min(scores[, 1:2]), max(scores[, 1:2])),
ylim=c(min(scores[, 1:2]), max(scores[, 1:2])), las=1)
```

The x and y limits of the plot are set to cover all the scores while keeping the same coordinates for each axis.

Labels are added to identify the samples, plotted at their scores.

```
> text(scores[, 1],scores[, 2], rownames(scores), col='gray',
cex=0.7)
```

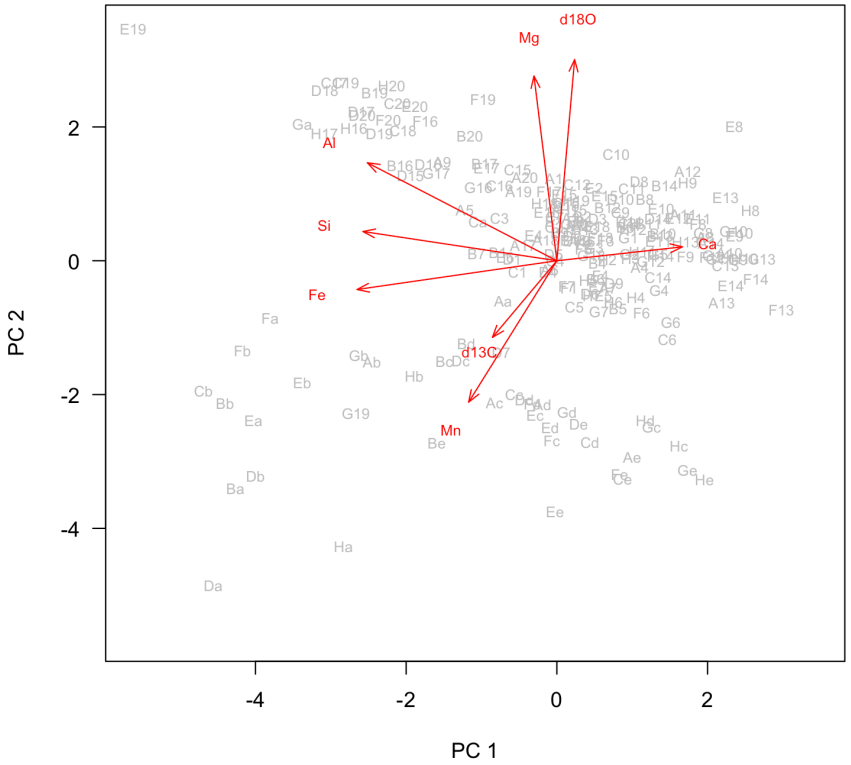
When there are many data points or the sample names are long, showing the sample names may make for an illegible plot, so it may be better in those cases to plot the sample scores as points.

Next, arrows are added to show the loadings of each variable on each axis. A scaling factor is added to make the arrows long enough to be visible, yet still keep them within the bounds of the plot. Some experimentation may be necessary for other data sets to get an appropriate scaling factor. The base of each arrow is at the origin (centroid), and the tip is at the loadings for the two principal components.

```
> scale <- 5
> arrows(0, 0, loadings[, 1]*scale, loadings[, 2]*scale,
length=0.1, angle=20, col='red')
```

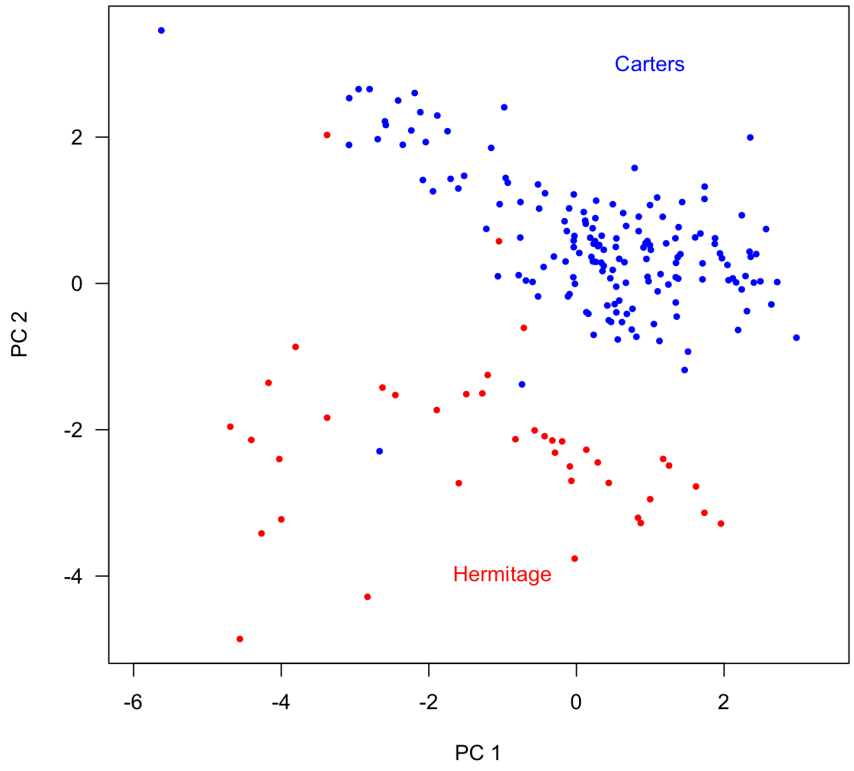
Labels are added to identify the loading vectors. Another scaling factor is added to get the labels to plot just beyond the arrowheads. Again, some experimentation with this factor may be needed for other data sets

```
> labelScale <- 1.2
> text(loadings[, 1]*scale*labelScale, loadings[, 2]*scale*
labelScale, rownames(loadings), col='red', cex=0.7)
```



The distribution of samples can be informative about underlying patterns. For example, there are two groups of data, a tight cloud in the upper right, and a more diffuse cloud towards the lower left. Often, color-coding the sample scores by some external data can reveal the origin of these patterns. Here, these two groups of samples correspond closely to those above and below an unconformity (a major interruption in sediment deposition). This surface lies at a stratigraphic position of 34.2 m, with rocks below belonging to the Carters Formation and rocks above being in the Hermitage Formation. Coding points this way is illuminating.

```
Carters <- nashville$StratPosition < 34.2
Hermitage <- nashville$StratPosition > 34.2
plot(scores[, 1], scores[, 2], xlab='PC 1', ylab='PC 2',
type='n', asp=1, las=1)
points(scores[Carters, 1], scores[Carters, 2], pch=16,
cex=0.7, col='blue')
points(scores[Hermitage, 1], scores[Hermitage, 2], pch=16,
cex=0.7, col='red')
text(1, 3, 'Carters', col='blue')
text(-1, -4, 'Hermitage', col='red')
```



6) One can also make a **correlation biplot**, which preserves the correlations among the variables (expressed by the angles between the loading vectors), albeit with some distortion of the distances among the samples. In this case, the coordinates are scaled by the standard deviations for each principal component (that is, the square root of the eigenvalue).

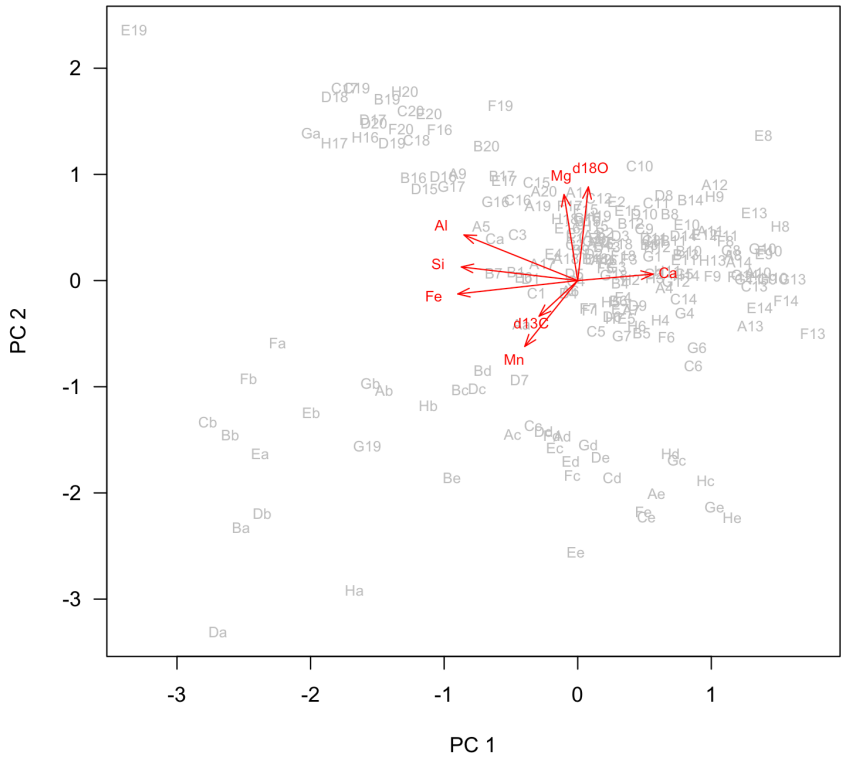
```
> sd <- pca$sdev
> plot(scores[, 1]/sd[1], scores[, 2]/sd[2], xlab='PC 1',
ylab='PC 2', type='n', las=1)
```

Labels for the sample scores are then added, again scaled by the standard deviations of the principal components.

```
> text(scores[, 1]/sd[1],scores[, 2]/sd[2], rownames(scores),
col='gray', cex=0.7)
```

Arrows are added to show the loadings, as are labels to identify the loadings. The coordinates of both are scaled by the standard deviations of the displayed principal components. The labels also need a scaling factor to get them to plot just beyond the arrowheads.

```
> arrows(0, 0, loadings[, 1]*sd[1], loadings[, 2]*sd[2],
length=0.1, angle=20, col='red')
> labelScale <- 1.2
> text(loadings[, 1]*sd[1]* labelScale, loadings[, 2]*sd[2]*
labelScale, rownames(loadings), col='red', cex=0.7)
```



The built-in **`biplot()`** function purports to create a correlation biplot, but it doesn't quite match the correlation biplot made following the description in Legendre and Legendre (1998). In addition, note that only the top and right axes match the coordinates of the sample scores. It is unclear what the built-in **`biplot()`** function is doing in this case, and I have been unable to replicate its plot using my own plotting commands.

```
> biplot(pca)
```

7) Although the loadings describe how each variable contributes to each principal component, one can also calculate the correlation coefficient between the variables and the principal components by transposing the loadings and multiplying by the standard deviations (square roots of the eigenvalues) of the principal components.

```
> sd <- pca$sdev
> correlations <- t(loadings) * sd
```

This can also be done by calculating the Pearson correlation coefficients of the sample scores and the original data.

```
> correlations <- cor(scores, geochem)
```

What to report

When reporting a principal components analysis, always include at least these items:

• A description of any data culling or data transformations that were used prior to ordination. State these in the order that they were performed.

• Whether the PCA was based on a variance-covariance matrix (i.e., **`scale.=FALSE`** was used) or a correlation matrix (i.e., **`scale.=TRUE`** was used).

• A scree plot that shows the explained variance of each of the principal components and that illustrates the criteria used for selecting the number of principal components to be studied.

• A table of loadings of all variables for each of the principal components that was studied. The table should highlight (e.g., with boldface) those loadings that are considered the most important for each principal component.

• One or more plots of sample scores that emphasizes the interpretation of the principal components, such as color-coding samples by an external variable. It is often useful to show the vectors corresponding to the loadings on these plots.

The deep dive: PCA by matrix operations

For those that are interested in matrix algebra that underlies a principal components analysis, it is possible to achieve the same solution as **`prcomp()`** purely through matrix operations in R.

First, create a correlation matrix.

```
> corMatrix <- cor(geochem)
```

Next, find the eigenvalues and eigenvectors of this correlation matrix through spectral decomposition.

```
> eigenObject <- eigen(corMatrix)
```

The **eigenObject** holds the eigenvalues (the variance explained by each principal component) in **eigenObject$values** and the eigenvectors (the loadings) in **eigenObject$vectors**. The loadings lack row and column labels, so it is helpful to add them.

```
> pcaVariances <- eigenObject$values
```

```
> pcaLoadings <- eigenObject$vectors
> rownames(pcaLoadings) <- colnames(geochem)
> PClabels <- paste('PC', 1:ncol(pcaLoadings), sep='')
> colnames(pcaLoadings) <- PClabels
```

Calculating the sample scores is done in three steps. First, we need to standardize our data so that the centroid is at the origin and all of the variables have the same variance, in other words, make all variables have a mean of zero and a standard deviation (variance) of 1. To do this, we create a function that will standardize one variable, and then we apply that function to every column (variable) in the data.

```
> standardize <- function(x) { (x - mean(x))/sd(x) }
> geochemStand <- apply(geochem, MARGIN=2, FUN=standardize)
```

Finally, sample scores are calculated by multiplying these standardized data by the loadings.

```
> pcaScores <- geochemStand %*% eigenObject$vectors
> colnames(pcaScores) <- PClabels
```

The scores, loadings, and variances are all identical to what the **prcomp()** function provides. You can check this by comparing **scores** to **pcaScores**, **loadings** to **pcaLoadings**, and **variances** to **pcaVariances**.

References

Legendre, P., and L. Legendre, 1998. Numerical Ecology. Elsevier: Amsterdam, 853 p.

Swan, A.R.H., and M. Sandilands, 1995. Introduction to Geological Data Analysis. Blackwell Science: Oxford, 446 p.

Theiling, B. P., L. B. Railsback, S. M. Holland, and D. E. Crowe, 2007. Heterogeneity in geochemical expression of subaerial exposure in limestones, and its implications for sampling to detect exposure surfaces. Journal of Sedimentary Research 77:159–169.