PRINCIPAL COMPONENTS ANALYSIS (PCA)

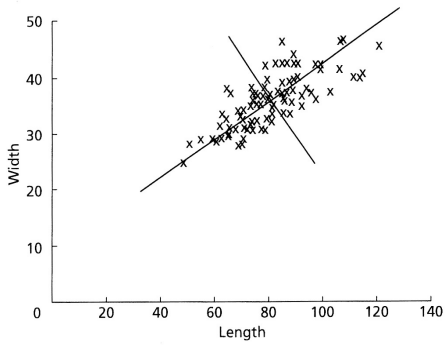
Steven M. Holland

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

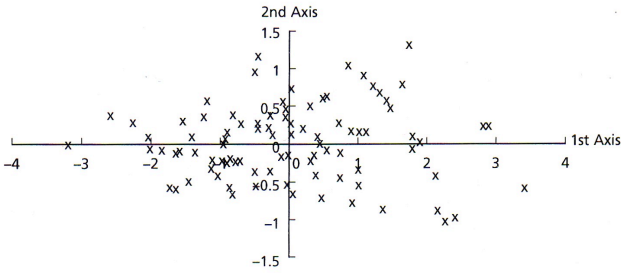
May 2008

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 replot 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

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

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 the constraint that their sum of squares is 1.

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

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.

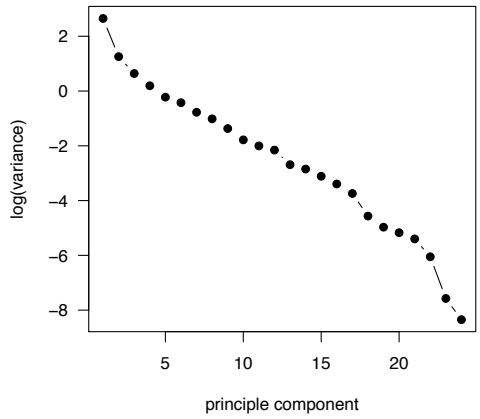
$$Y_2 = a_{21}X_1 + a_{22}X_2 + ... + a_{2p}X_p$$

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

$$Y = AX$$

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 Sx, 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 Sy, 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

$$Y_{kr} = a_{k1}x_{k1} + a_{k2}x_{k2} + ... + a_{kp}x_{kp}$$

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

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

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) Do an R-mode PCA using prcomp() in R. To do a Q-mode PCA, the data set should be transposed before proceeding. R-mode PCA examines the correlations or covariances among variables, whereas Q-mode focusses on the correlations or covariances among samples.

```
> mydata <- read.table(file="mydata.txt", header=TRUE,
row.names=1, sep=",")

> mydata.pca <- prcomp(mydata, retx=TRUE, center=TRUE,
scale.=TRUE)
# variable means set to zero, and variances set to one
# sample scores stored in mydata.pca$x
# loadings stored in mydata.pca$rotation
# singular values (square roots of eigenvalues) stored
#  in mydata.pca$sdev
# variable means stored in mydata.pca$center
# variable standard deviations stored in mydata.pca$scale

> sd <- mydata.pca$sdev
> loadings <- mydata.pca$rotation
> rownames(loadings) <- colnames(mydata)
> scores <- mydata.pca$x
```

2) Do a PCA longhand in R:

```
> R <- cor(mydata)
# calculate a correlation matrix

> myEig <- eigen(R)
# find the eigenvalues and eigenvectors of correlation matrix
# eigenvalues stored in myEig$values
# eigenvectors (loadings) stored in myEig$vectors

> sdLONG <- sqrt(myEig$values)
# calculating singular values from eigenvalues

> loadingsLONG <- myEig$vectors
```

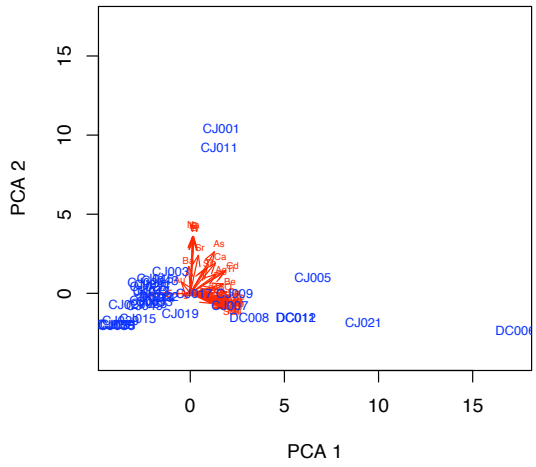
```
> rownames(loadingsLONG) <- colnames(mydata)
# saving as loadings, and setting rownames

> standardize <- function(x) {(x - mean(x))/sd(x)}
> X <- apply(mydata, MARGIN=2, FUN=standardize)
# transforming data to zero mean and unit variance

> scoresLONG <- X %*% loadingsLONG
# calculating scores from eigenanalysis results
```

3) Compare results from the two analyses to demonstrate equivalency. Maximum differences should be close to zero if the two approaches are equivalent.

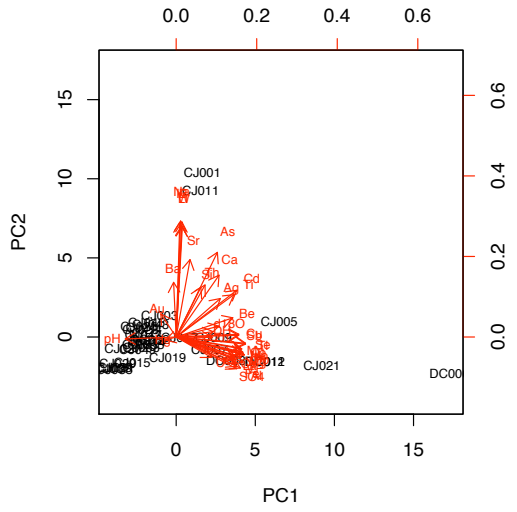
```
> range(sd - sdLONG)
> range(loadings - loadingsLONG)
> range(scores - scoresLONG)
```

4) Do a distance biplot (see Legendre & Legendre, 1998, p. 403)

```
> quartz(height=7, width=7)
> plot(scores[,1], scores[,2], xlab="PCA 1", ylab="PCA 2",
   type="n", xlim=c(min(scores[,1:2]), max(scores[,1:2])),
   ylim=c(min(scores[,1:2]), max(scores[,1:2])))
> arrows(0,0,loadings[,1]*10,loadings[,2]*10, length=0.1,
   angle=20, col="red")
# note that this scaling factor of 10 may need to be changed,
# depending on the data set

> text(loadings[,1]*10*1.2,loadings[,2]*10*1.2,
   rownames(loadings), col="red", cex=0.7)
# 1.2 scaling insures that labels are plotted just beyond
# the arrows

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



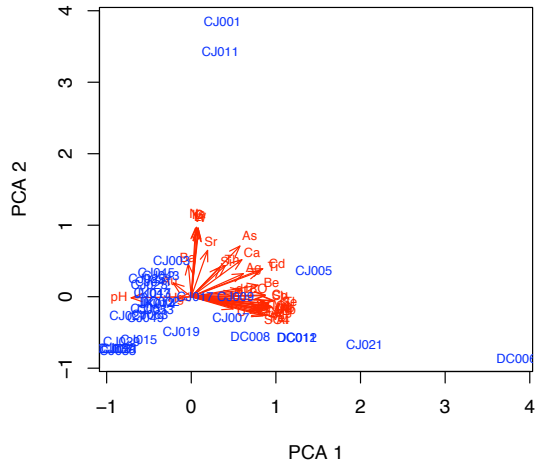

```
> quartz(height=7, width=7)
> biplot(scores[,1:2], loadings[,1:2], xlab=rownames(scores),
   ylab=rownames(loadings), cex=0.7)
# using built-in biplot function
```



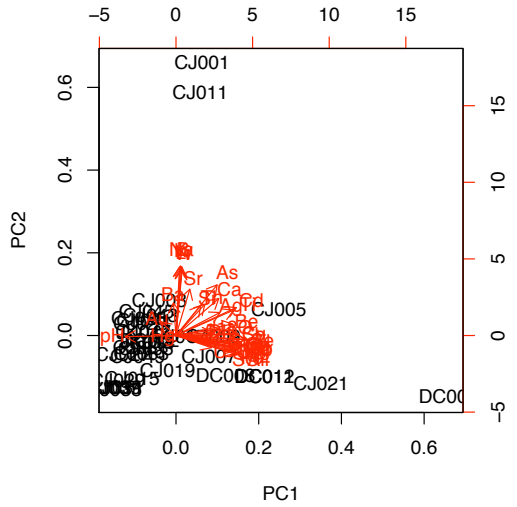
5) Do a correlation biplot (see Legendre & Legendre, 1998, p. 404)

```
> quartz(height=7, width=7)
> plot(scores[,1]/sd[1], scores[,2]/sd[2], xlab="PCA 1",
   ylab="PCA 2", type="n")
> arrows(0,0,loadings[,1]*sd[1],loadings[,2]*sd[2],
```

```
   length=0.1, angle=20, col="red")
> text(loadings[,1]*sd[1]*1.2,loadings[,2]*sd[2]*1.2,
   rownames(loadings), col="red", cex=0.7)
> text(scores[,1]/sd[1],scores[,2]/sd[2], rownames(scores),
   col="blue", cex=0.7)
# 1.2 scaling insures that labels are plotted just beyond
# the arrows
```




```
> quartz(height=7, width=7)
> biplot(mydata.pca)
# using built-in biplot function of prcomp(); note that only
# the top and right axes match the coordinates of the points;
# also note that still doesn't quite replicate the correlation
# biplot.  It's unclear what this function really does.
```

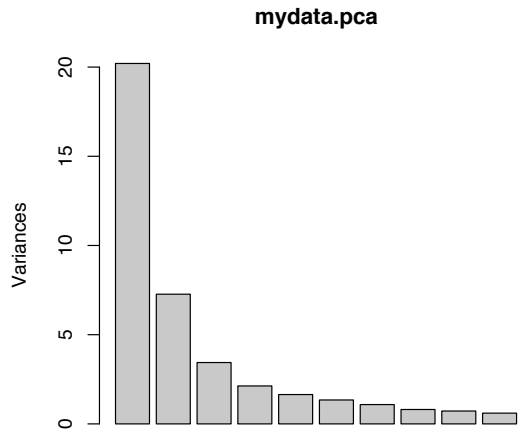


6) Calculate the correlation coefficients between variables and principal components

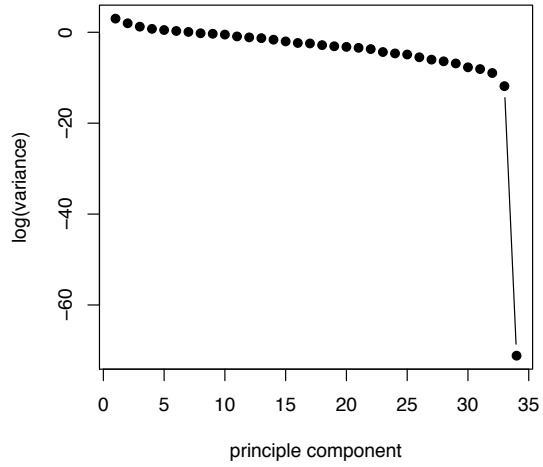
```
> correlations <- t(loadings)*sd
# find the correlation of all the variables with all PC's

> correlations <- cor(scores,mydata)
# another way to find these correlations
```

7) Plot a scree graph

```
> quartz(height=7, width=7)
> plot(mydata.pca)
# using built-in function for prcomp; may not show all PCs
```




```
> quartz(height=7, width=7)
> plot(log(sd^2), xlab="principle component",
   ylab="log(variance)", type="b", pch=16)
# using a general plot, with variance on a log scale
```



8) Find variance along each principal component and the eigenvalues

```
> newsd <- sd(scores)
> max (sd - newsd)
# finds maximum difference between the standard deviation form
```

```
# prcomp and the standard deviation calculated longhand;
# should be close to zero

> eigenvalues <- sd^2
> sum(eigenvalues)
# should equal number of variables

> length(mydata)
# number of variables
```

9) Save loadings, scores, and singular values to files

```
> write.table(loadings, file="loadings.txt")
> write.table(scores, file="scores.txt")
> write.table(sd, file="sd.txt")
```

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.