Accepted author manuscript version reprinted, by permission, from International Journal of Sports Physiology & Performance Oct 2019, Vol. 14 Issue 9, pp.1304-1310, https://doi.org/10.1123/ijspp.2019-0045. © Human Kinetics, Inc

Supplementary Information

Principal component analysis (PCA) is not a statistical technique *per-se*, rather it is a linear algebra technique, based on eigen-decomposition of the covariance matrix of the data, which is typically used to reduce higher-dimensional data (i.e. data containing many variables) into a lower-dimensional state (i.e. data containing just a few variables) without any great loss of information. As such, PCA is primarily a dimension reduction technique, which allows the user to capture the complexity of a data system in just a few uncorrelated (orthogonal) composite variables known as principal components (PCs). Individual PCs are constructed using linear weighted combinations of the original measured variables, with the weighted coefficients for the respective PCs contained in a matrix of eigenvectors. These eigenvectors, together with the corresponding eigenvalues, are automatically generated when eigen-decomposition is performed, with the eigenvectors corresponding to the largest eigenvalues accounting for most of the variance in the data. Because it is possible to capture most of the variance in the data in the first two or three orthogonal PCs, this means that complex higher-dimensional data systems can be represented on 2D and 3D scatter plots with minimal loss of information, making PCA an ideal tool for visualizing complex data. Furthermore, because the new variables are orthogonal it means that they are not correlated in any way, thus ensuring that the individual PCs capture different attributes within the data system.

In order to perform PCA, a ($n \times m$) matrix, X, should be created containing the data to be analysed. The columns of the X matrix comprise the variables (which should be zero mean-centred and generally standardized to unit variance to produce z-scores), while the respective rows represent individual subjects or observations. From this the covariance matrix, C, can be computed as follows (where T is the transpose):

$$C = X^T X \qquad \text{(S1)}$$

Eigen-decomposition of the covariance matrix, C, should then be performed to compute the matrix of eigenvalues, D, and the matrix of eigenvectors, V, as follows:

$$C = VDV^T \qquad \text{(S2)}$$

Finally, the original standardised data should be projected into the eigenspace of the covariance matrix to compute matrix of principal components, PC, as follows:

$$PC = XV \qquad \text{(S3)}$$

Generic principal component analysis (PCA) R code

The following code is generic and can be used in any application to perform simple PCA. In order to use it, simply copy and paste the code into ‘R’ or ‘RStudio’.

```
# Create PCA model using the ‘prcomp’ function from the 'stats' package in R.
# Here 'data' refers to the raw data that is to be analysed, and the commands 'center' and 'scale.' tell
# the 'prcomp' function that the data should be zero mean centred and standardized to unit variance
# (i.e. transforming the data to Z-scores).

pca.model <- prcomp(data, center = TRUE, scale. = TRUE)
summary(pca.model)        # This displays the variance attributable to each principal component.
pca.model$sdev            # This displays the normalized singular values.
(pca.model.eigvals <- pca.model$sdev^2)        # These are the normalized eigenvalues.
pca.model$rotation        # These are the eigenvectors.
pca.model$x               # These are the principal components.

# Produce a scatter plot of the scores for the first two principal components.
PC1 <- pca.model$x[,1]
PC2 <- pca.model$x[,2]
plot(PC1,PC2, main = "Scatter plot of PCs", pch=20, col="red", xlab = "First PC", ylab = "Second PC")
abline(h=0, lty=2)
abline(v=0, lty=2)

# Use the 'biplot' command to produce simple biplot of the first and second principal components.
biplot(pca.model, scale=0)
```