Principal Component Analysis:  
An Introduction with Examples in R

*Salvatore A. Sidoti | PhD Candidate | Behavioral Ecology |*

*The Department of Evolution, Ecology & Organismal Biology |*[*sidoti.23@osu.edu*](mailto:sidoti.23@osu.edu)

*2019-02-11*

*The HTML version of this discussion is featured on* [*Rpubs*](http://www.rpubs.com/carabidus/465971)

Introduction

Principle component analysis (PCA) distributes the variation in a multivariate dataset across *components*. This allows us to visualize patterns that would not be apparent with common graphical techniques. Linear algebra is at the heart of the PCA, but this discussion will be light on mathematical theory. Instead, you can expect a gentle introduction to the topic, which will include how this ordination technique in carried out in R.

Accomplishing the PCA *Manually*

With the powerful tools available to us in R, there is no need to conduct a PCA manually. Contained within one line of code, R has native functions which can handle the heavy-lifting for us. My goal for the *manual* PCA is to expose you to the terminology and concepts in PCA. As such, you will be better prepared to defend your analysis.

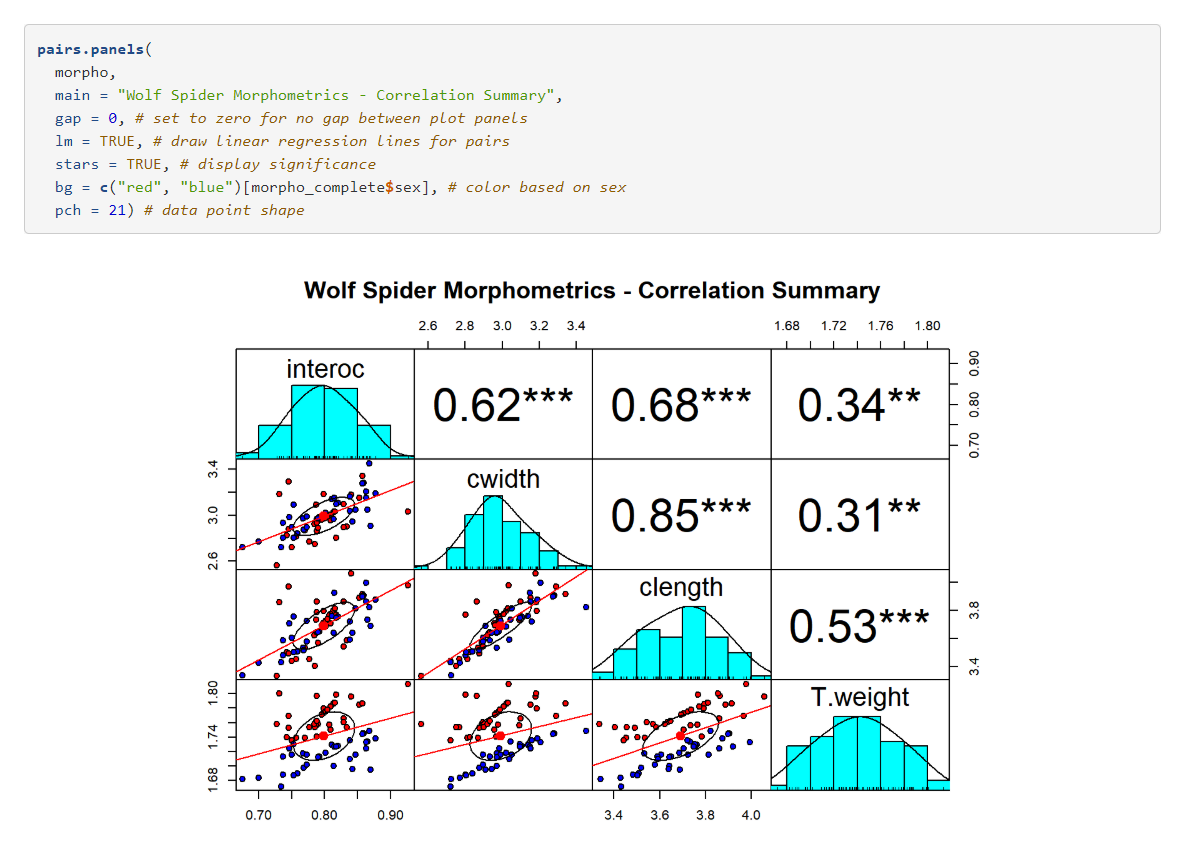
Motivating example - wolf spider morphometrics

The original motivation for this analysis was to establish a standard algorithm to determine the “size” of a wolf spider. One way to accomplish this is with a PCA of morphometric characteristics. The parameters which possess the highest degree of variation will be the most optimal predictor of animal size.



Blue: interocular distance, Green: carapace width, Red: carapace length





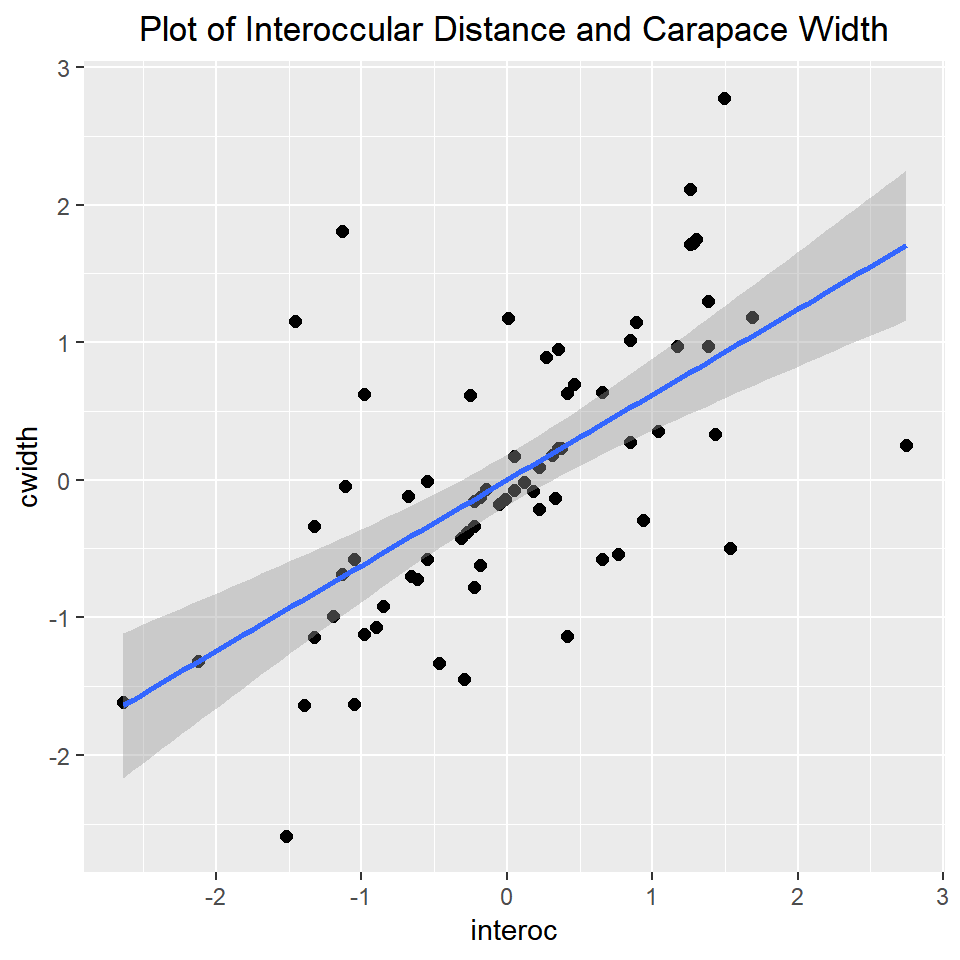
## Covariance or Correlation?

PCA is a dimensionality reduction technique that allows us to see latent patterns in the data. To do this, the PCA is based heavily on concepts in linear algebra: eigenvalues and eigenvectors are at the heart of the PCA. First, we need to establish whether the metrics in our dataset are like or mixed. If the dataset contains the same units of measure across all of the variables – i.e. all variables are weights in grams – we standardize the data via mean-centering and employ a covariance matrix. However, if our data contains mixed units – like the morphometric data in this example – we mean-center the data, divide by the standard deviation, and employ a correlation matrix. Which matrix you choose becomes important when setting the parameters of the built-in PCA functions in R.

Our data has mixed metrics - weight (mg) and linear measures (mm).

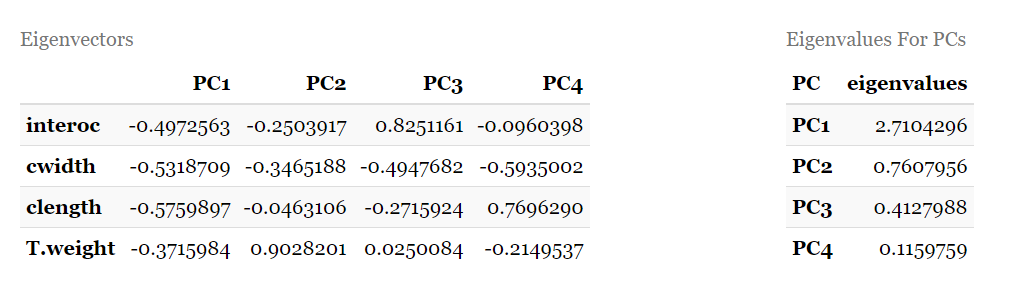
## Standardize the Data: Mean-center and Divide by the Standard Deviation





## Find the Eigenvalues & Eigenvectors





Each column in the table above(left) is an eigenvector. Each eigenvector is a principal component (PC) with its own eigenvalue. These eigenvalues are given in table 2. The eigenvalues are used to identify which principal component(s) have the strongest correlation with the overall dataset: the higher the eigenvalue, the stronger its correlation. Each eigenvector is a normalized linear combination of the variables interoc, cwidth, clength, T.weight.

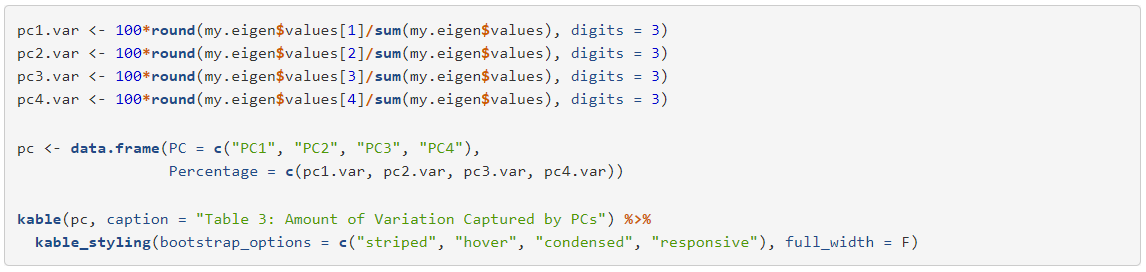
The coefficients are known as loadings which are values [-1, 1]. We can see that the variables of interest in PC1 trend together due to all of the coefficients having the same sign.

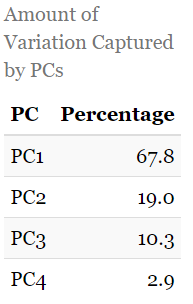
Note that the sum of the eigenvalues equals the total variance of the scaled data.



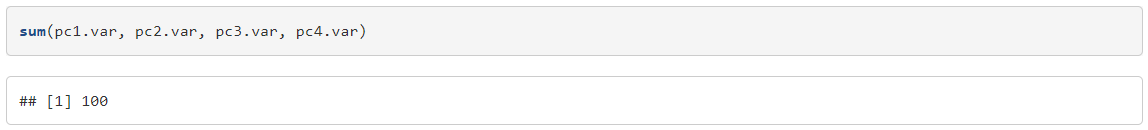
## Amount of Variation Captured by the PCs in the Dataset

Take each eigenvalue and divide it by the sum of all eigenvalues – the total variation in the dataset – then multiply by 100. The following yields a table with the percent variation for each PC.



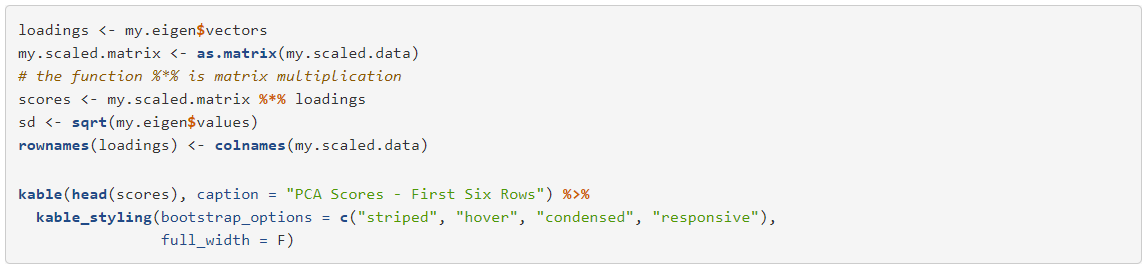


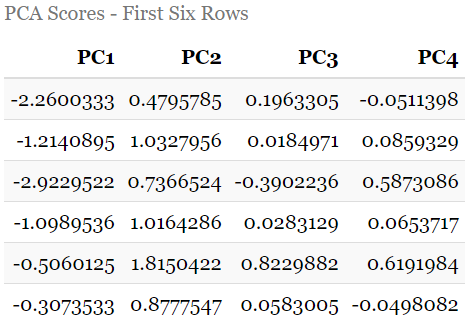
The total variation should sum to approximately 100% depending on rounding error:



## What are the PCA “scores”?

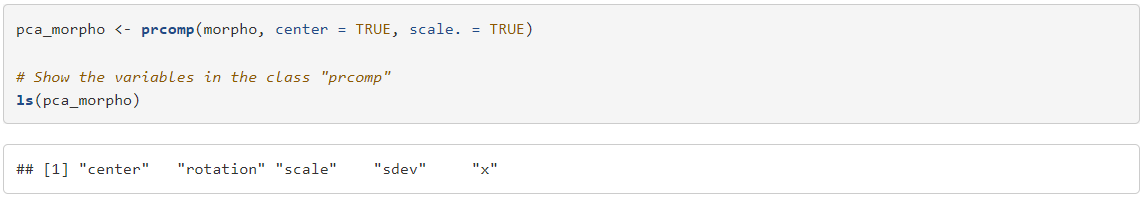
Express the loadings and scaled data as matrices, then multiply them together. The result is a new matrix which expresses the data in terms of the PCs. These are the PCA scores.





# Accomplishing the PCA with Native R Functions

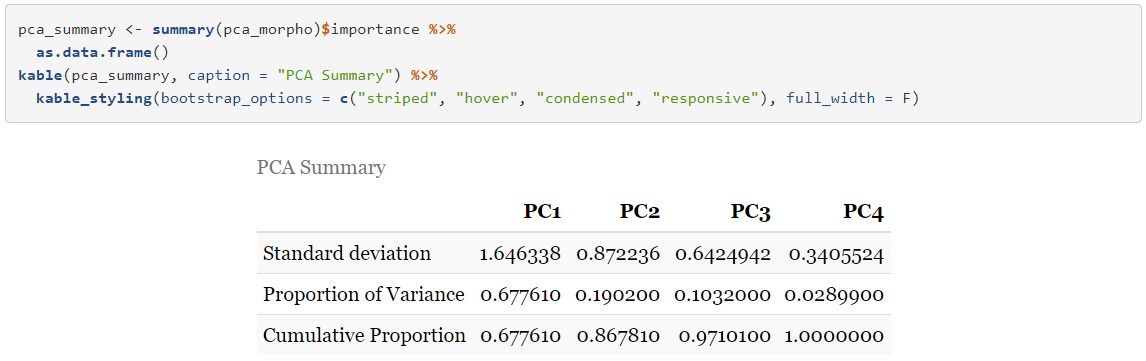
The function prcomp is the primary tool for PCA in base R.



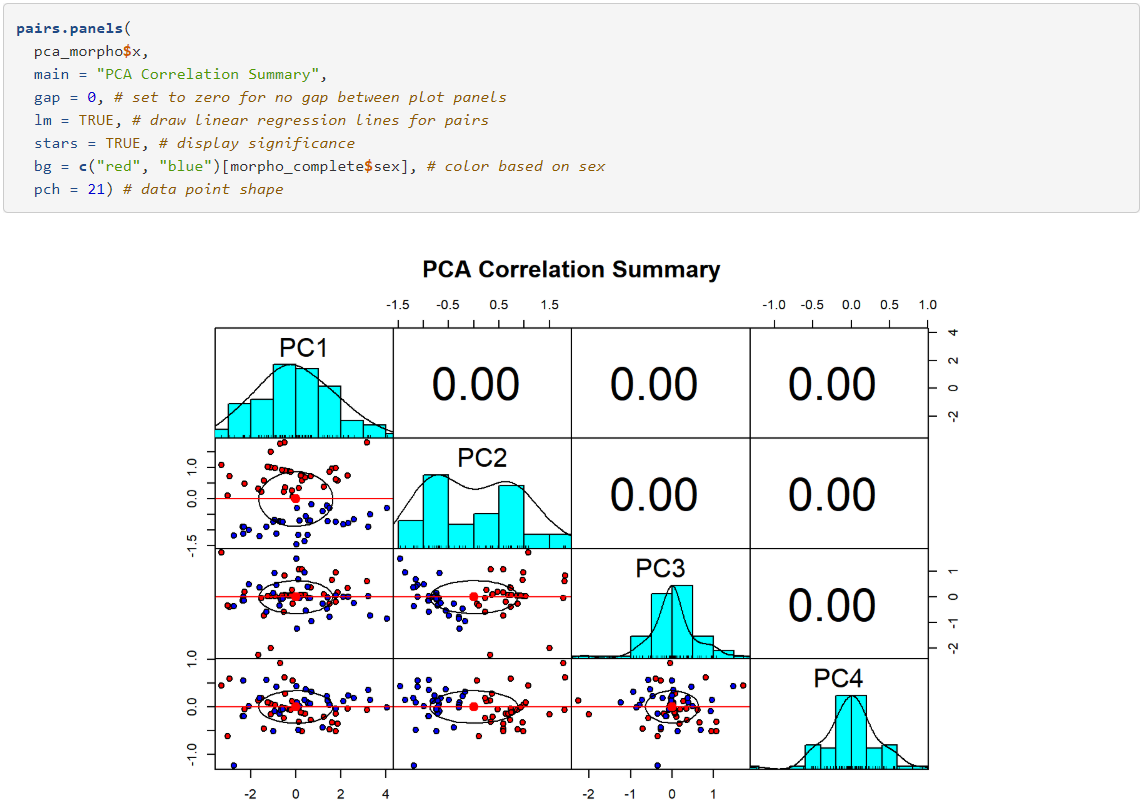
Rotations are often referred to as the “loadings” of a PCA.



Summary output of the PCA:



## Orthogonality of PCs

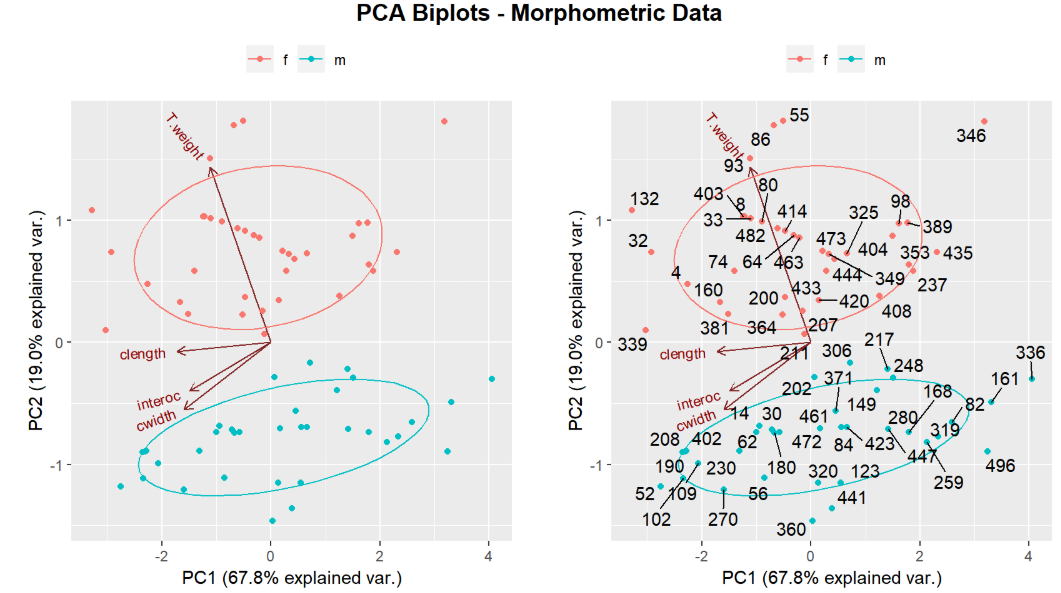


PCs are orthogonal vectors. Thus, the correlation coefficients equal zero for all possible pairwise combinations of PCs. Therefore, PCA is often used as a precursor to predictive modeling procedures as multicollinearity is eliminated.

## Biplot

The biplot is the primary visualization tool for PCA.



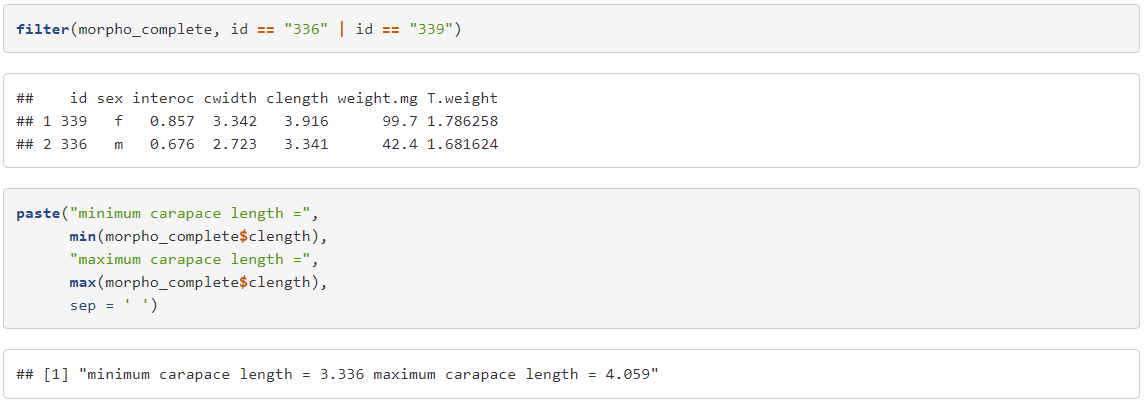


PCA biplots without and with spider ID labels. Pink: females, Red: males.

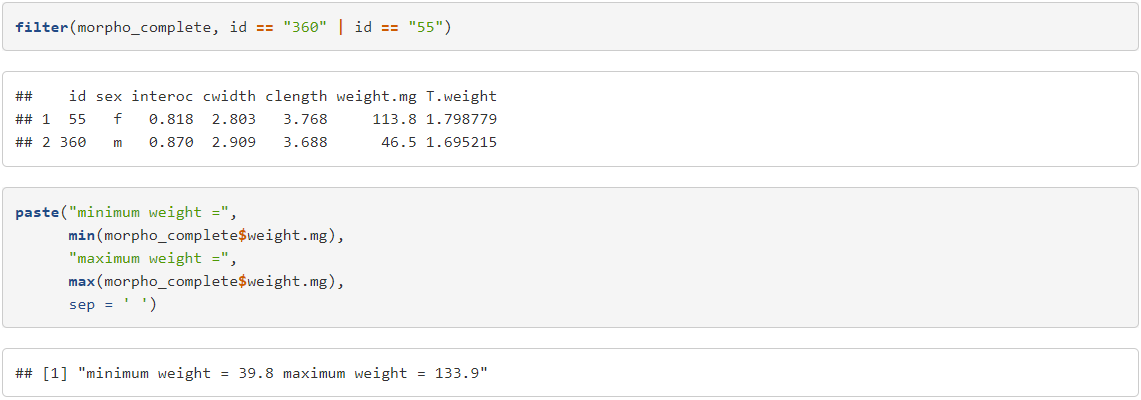
## Interpreting the Biplot

Each variable in the dataset is plotted as a vector (red arrows). The cosine of the angle between any two vectors equals their correlation. For instance interocular width and carapace width have a very small angle between them, indicating their high correlation.

Regarding carapace length, let’s look at spider ID#336 (male) and ID#339 (female):



Regarding weight, let’s look at spider ID#360 (male) and ID#55 (female):

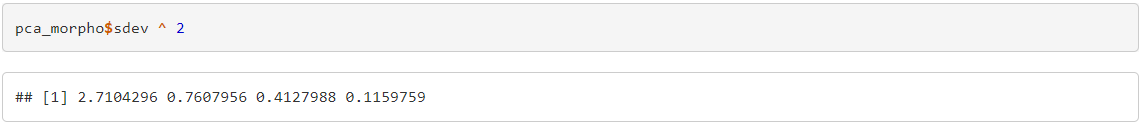


## How many PCs can I retain to exp-lain “enough” variation?

Many methods have been propsed to answer this, but two are the most common: the Kaiser Criterion and Parallel Analysis

### **Kaiser Criterion**

If an eigenvalue associated with a PC is >1>1, then you retain that component. To compute the eigenvalues from the PCA, square the standard deviations in the prcomp object.

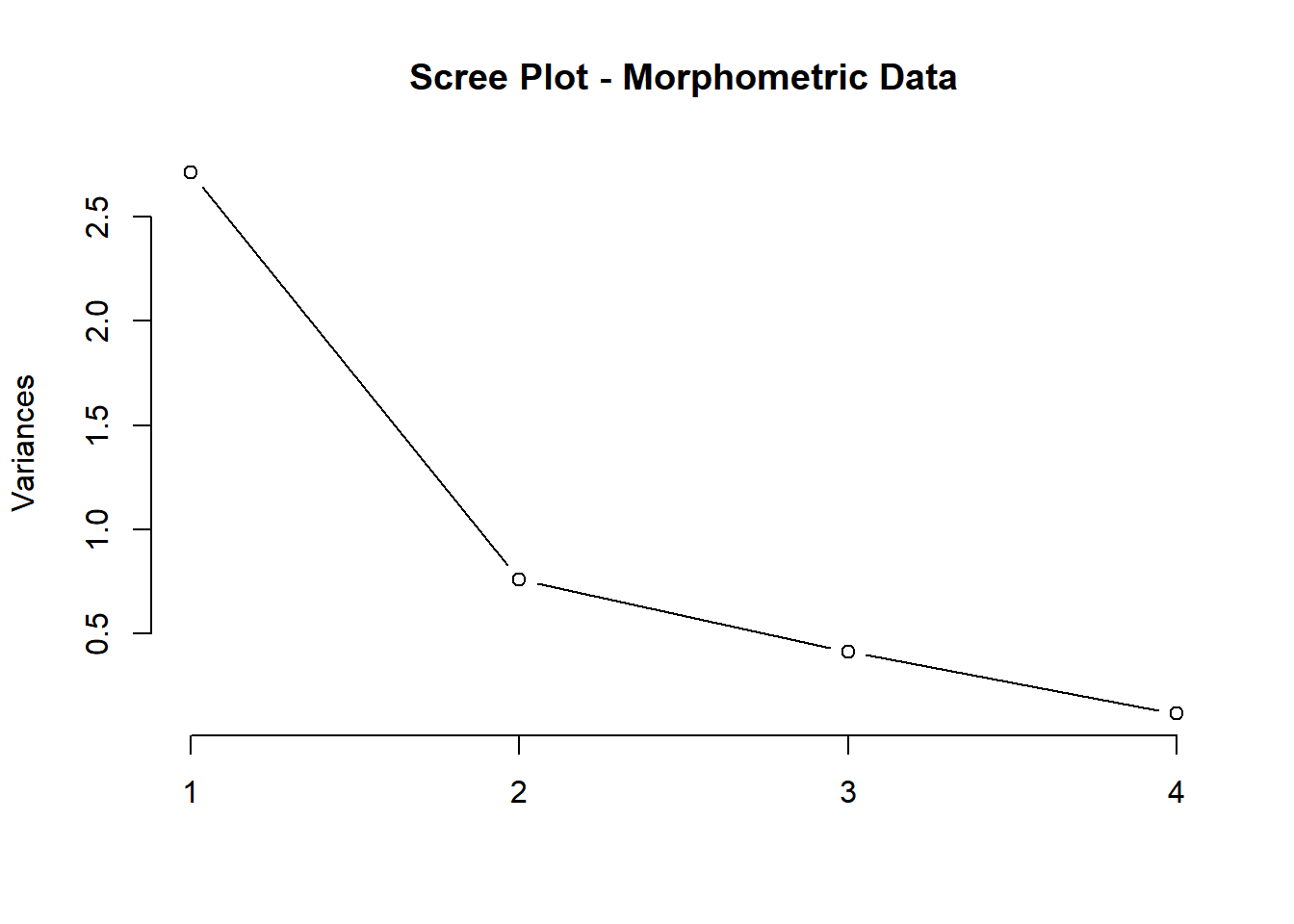


The reason this works: the sum of the eigenvalues is equal to the total variance in the dataset. Only the first eigenvalue is >1>1, so according to the Kaiser Criterion, PC1 is sufficient to explain the variation in the dataset.

### **Parallel Analysis**

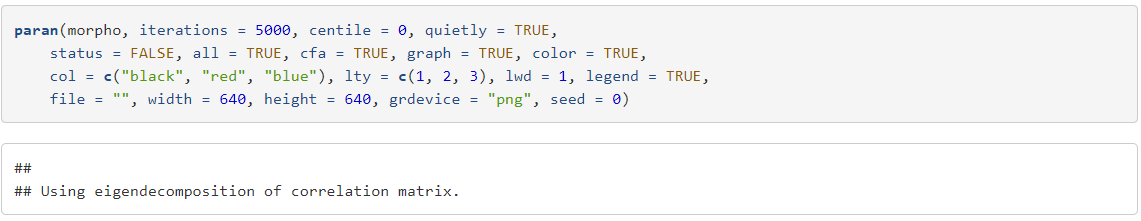
Parallel analysis is a technique designed to reduce the subjectivity of interpreting a scree plot.

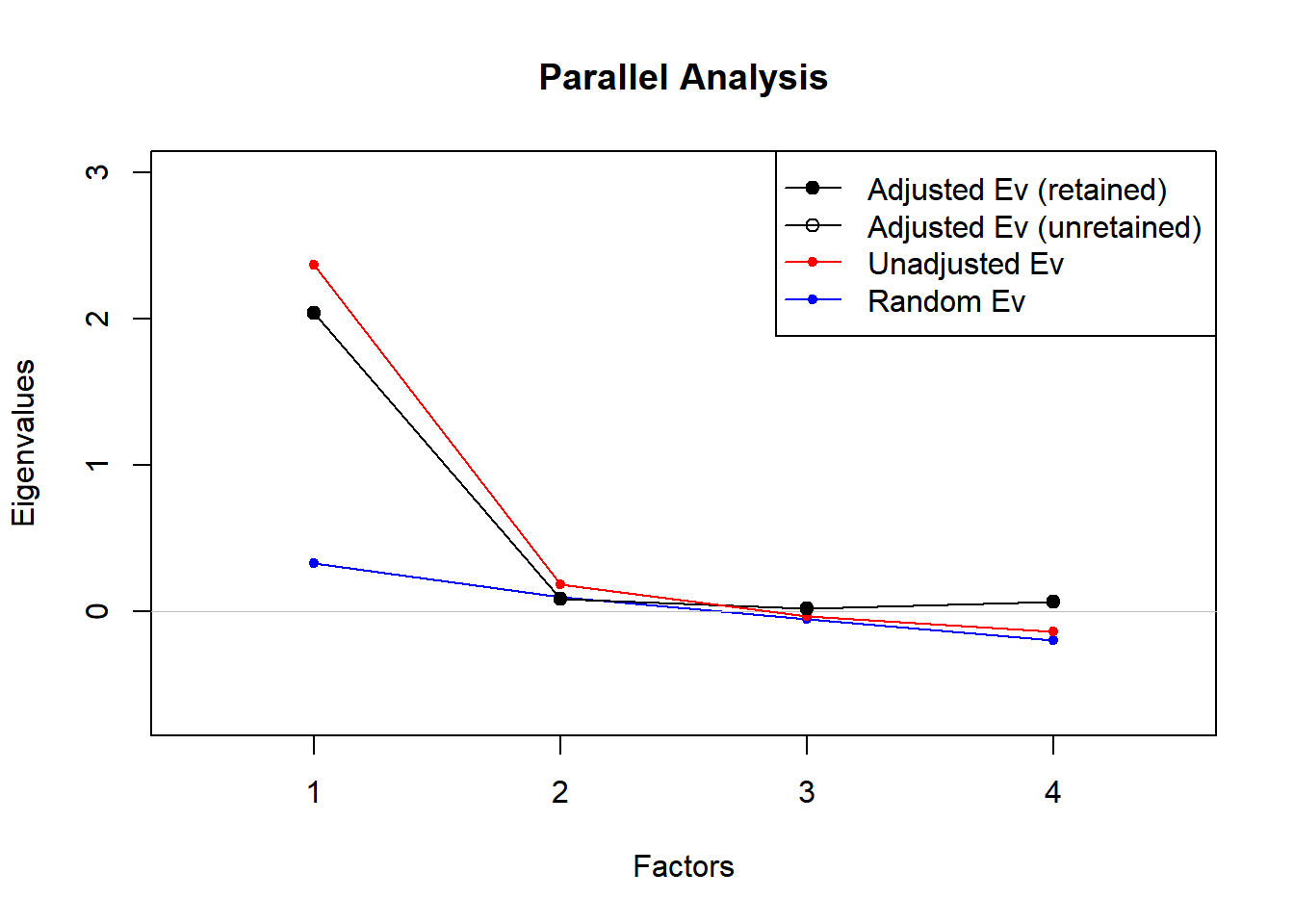




A common visual method is to choose the point which creates the most extreme “elbow” and retain that many PCs (x-axis). Accordingly, we would retain PC1 and PC2.

On the other hand, parallel analysis is a simulation-based method that generates thousands of data sets with the same number of items and range of the “real” dataset. Then, we retain the number of factors with observed eigenvalues larger than those extracted from the simulated data.



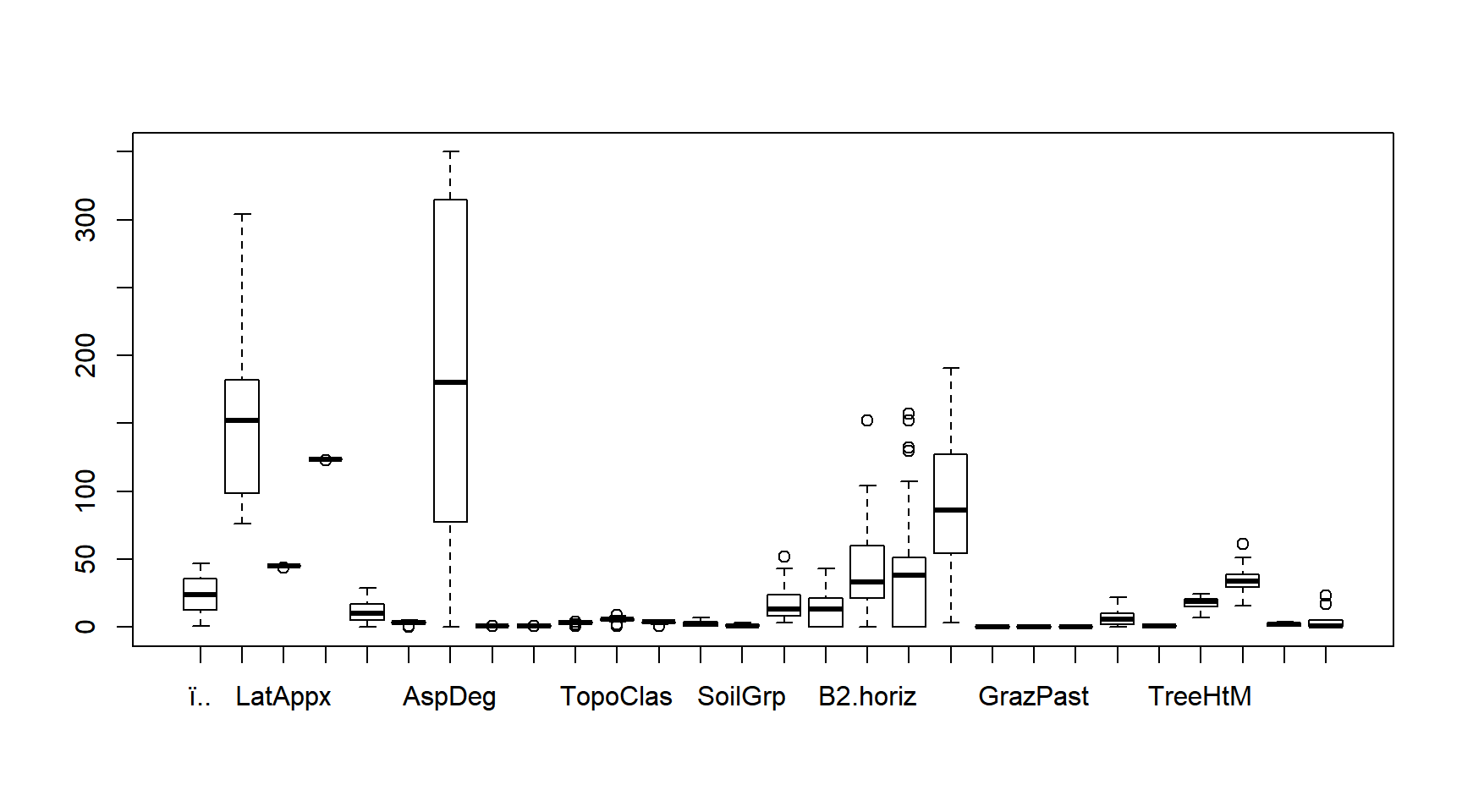


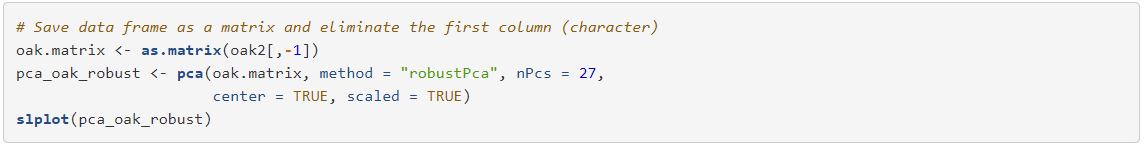
The results show that we would retain only PC1, which agrees with the Kaiser Criterion method.

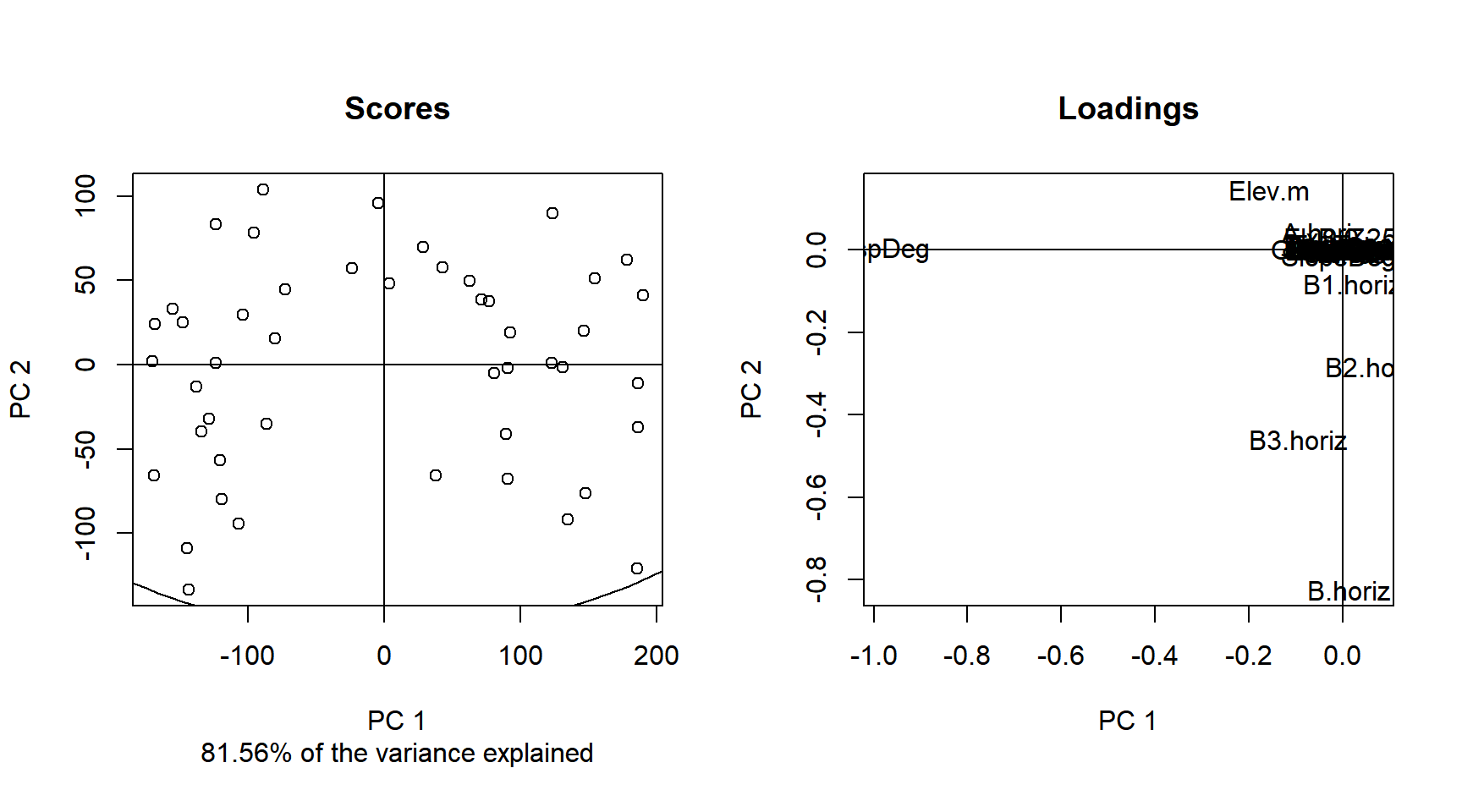
# PCA on the Oak Woods Dataset - Robust PCA

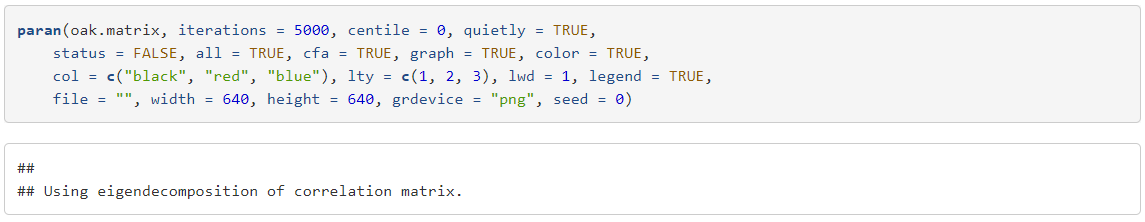
The morphometric dataset we used here is virtually “ideal” for running a generic PCA in R. It contains no missing values, zeroes, extreme values or outliers. However, the Oak Woods dataset is characterized by extreme variability, outliers and zero values:

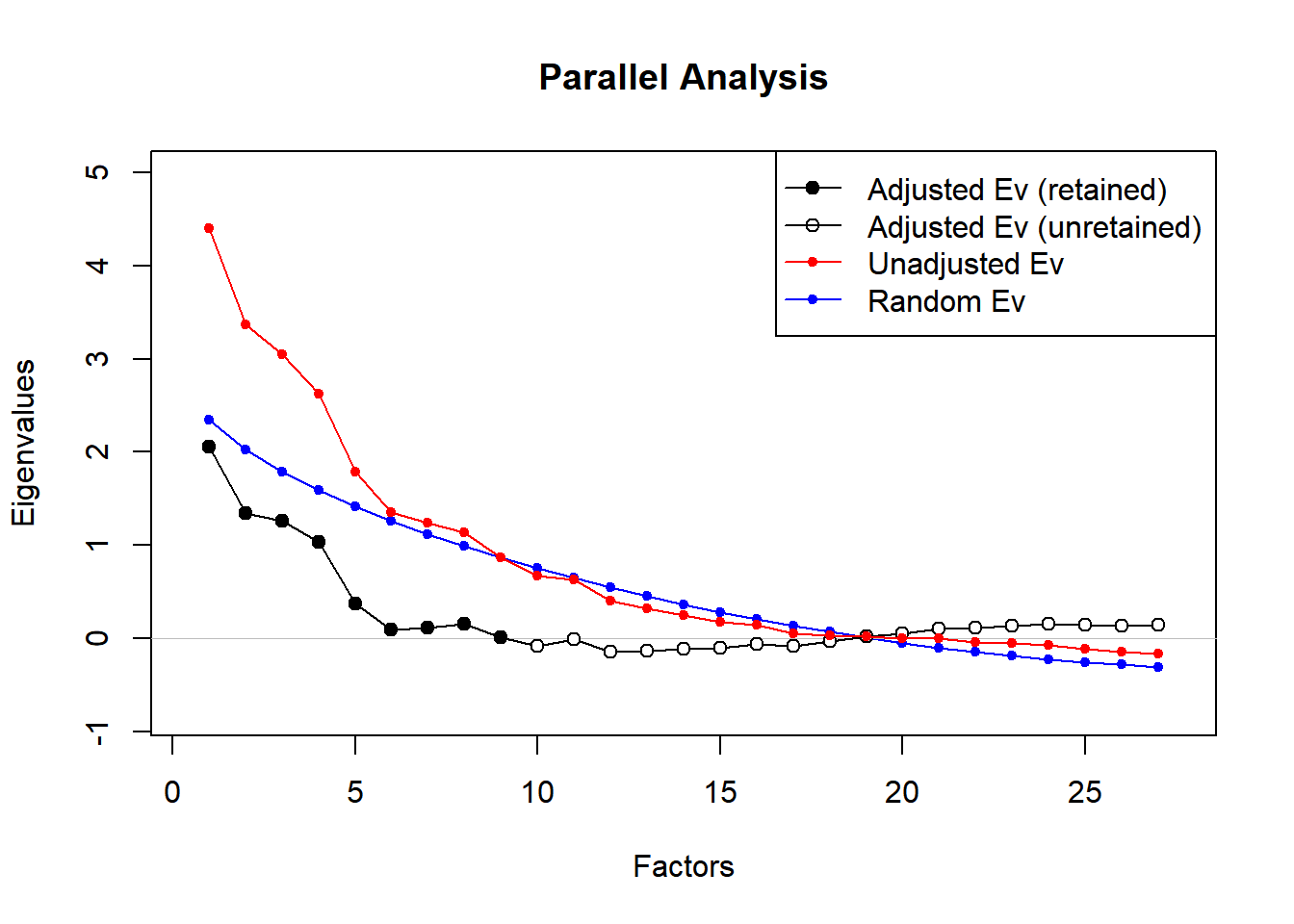


As such, a prudent course would be to conduct a robust PCA.









In Summary, a PCA of any type may not be an appropriate statistical approach for this dataset. PCoA is a better choice with its built-in flexibility with respect to distance methods. Also, clustering based on any number of variables, such as stand, Elev.m, or AspClass would likely yield more meaningful results.