6. Principal Component Analysis

FISH 560: Applied Multivariate Statistics for Ecologists

Topics

- Perform principal component analysis (PCA)

R Packages: vegan, FactoMineR, factoextra, dplyr
R Source: biostats

BACKGROUND

Ordination or scaling methods achieve an efficient and optimized low-dimensional representation of a complex data structure by emphasizing and bringing to the forefront underlying trended variation while suppressing "noise" (Kenkel 2006). This objective is readily achievable since multiple variables normally show coordinated responses to one or more (typically many) underlying factors. Ordination methods are generally used in exploratory data analysis, both to search for and summarize underlying trends, and to examine interrelationships among variables. A number of ordination approaches are available, including principal component analysis (PCA) and its variants, correspondence analysis (CA), and non-metric multidimensional scaling (NMDS). We will cover these approaches and others in the following chapters.

Principal component analysis (PCA) is an ordination technique that reduces the number of variables in an object-by-descriptor variable matrix into a fewer number of synthetic dimensions that are linear combinations, or principal components (PC), of the original variables. Put another way, PCA involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first axis maximizes linear variance (i.e., it summarizes the dominant linear trend), the second maximizes the residual variance not accounted for by the first axis (i.e., subdominant linear trend), and so forth. The method is somewhat analogous to simple linear regression, but extended to multiple dimensions and without distinguishing between dependent and independent variables. In general, the aim is to represent the objects (rows) and variables (columns) of the data set in a new system of coordinates in reduced space (e.g., two or three axes or dimensions). The resulting ordination plot (usually depicted using the first 2 or 3 PCs) is used to interpret object similarity: objects that are close in ordinate space share similar properties and those distant are increasingly dissimilar. Conceptually, ordination complements cluster analysis in that they both aid interpretation and highlight structure underlying object similarity using multivariate data. However, PCA also allows one to investigate which variables are the primary drivers of object dissimilarity by examining the loadings (correlations) of the original variables against each PC.

In practice, PCA is either performed on a variance–covariance matrix or on a correlation matrix. The *variance–covariance approach* is followed when the same units or data types are used. The aim is then to preserve and to represent the relative positions of the objects and the magnitude of variation between variables in the reduced space. By contrast, PCA on a *correlation matrix* is used when descriptor variables are measured in different units or on different scales (e.g. different environmental parameters) or when the aim is to display the correlations among (standardized) descriptor variables. The two approaches lead to different principal components and different distances between projected objects in the ordination; hence, the interpretation of the relationships must be made with care. Indeed, for correlation matrices, variables are first standardized (i.e. they become independent of their original scales), and so distances between objects are also independent from the scales of the original variables. All variables thus contribute to the same extent to the ordination of objects, regardless of their original variance.

The mathematical basis of PCA is similar to multiple regression analysis, and correspondingly, requires that the original variables meet basic assumptions of multivariate normality, homogeneity in variance and several others which are documented elsewhere (Legendre and Legendre 1998) and discussed previously in class. As an aside, these assumptions are generally violated by species abundance data that are often Poisson or log-normally distributed. Transformation of species abundances can sometimes abate violations of normality, but PCA also operates under the assumption that the original variables are represented in Euclidean distance. The high frequency of zeros that are typical of species community datasets makes this a dubious assumption. The resulting ordination space may be distorted and provide little ecologically relevant information on object similarity when using species data. On the other hand, continuous environmental data such as temperature, salinity, etc., are normally distributed (or transformations thereof) and are ideal for PCA.

SET-UP

In this exercise you will be working with the MAHA environment dataset. Import the dataset by typing:

```
envdata <- read.csv('MAHA_environment.csv', header=TRUE, row.names=1)
```

For exploratory purposes, we will also be using the MAHA site group dataset that contains three column vectors depicting the major basin, degree of human disturbance and a column of text colors (which will help for graphing later) for each site (object). Import the dataset by typing:

```
sitegroup <- read.csv('MAHA_Groups.csv',header=TRUE, row.names=1)
```

Let's first set-up your R work session by defining the current work directory to your folder of choice and loading the Packages listed at the beginning of this chapter. Also, make sure to source the BIOSTATS file from the *File* pull-down menu. You can also do this using the functions **`setwd`**, **`library`** and **`source`**.

PRINCIPAL COMPONENT ANALYSIS

There are two basic functions for performing PCA in R: princomp() and prcomp(). Essentially, they compute the same values (technically, princomp computes an eigenanalysis and prcomp computes a singular value decomposition). We will make use of the prcomp () function.

Conduct the PCA

The prcomp() function accepts matrices or data frames that contain only the samples and variables of interest. Alternatively, you can select the columns from a larger matrix to include the analysis. The prcomp() function allows us to specify a number of parameters concerning the calculation. The most important is whether we want to use a correlation or variance-covariance matrix (see lecture and discussion above). To reiterate, PCA is sensitive to the scale of measurement of the 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). In prcomp(), this means specifying scale=TRUE in the function call.

The usage of prcomp() is:

prcomp(x, scale=FALSE, scores=TRUE)

Where

- **`x`** is the object×variable matrix
- **`scale`** is a logical value. If TRUE, the original variables are scaled to unit variance (this is the correlation matrix and is almost always preferred). Default is FALSE and therefore the variance-covariance matrix is used in the PCA.

- **scores** is a logical value. If TRUE, PC scores are retained in the resulting PCA object. Default is TRUE.

For additional documentation of all the arguments type **?prcomp**.

Let's perform PCA by typing:

```
env.pca<-prcomp(envdata, scale=TRUE)
```

Interpreting the results from the PCA

The results of a PCA are complex and stored in a list, including:

- variance (eigenvalue) explained by each eigenvector
- variable loadings by eigenvector
- sample scores and ordination plots

First, let's evaluate how much variance in the dataset was captured by the resulting principal components (PCs). We do this by calling the summary of the object env.pca to view the variance explained by each PC and the cumulative variation explained sequentially (the results for PC 9 and 10 are omitted for clarity). Do this by typing:

```
summary(env.pca)
```

```
Importance of components:
                         PC1   PC2   PC3   PC4    PC5    PC6    PC7    PC8
Standard deviation     1.655 1.377 1.203 1.168 0.8916 0.7472 0.7225 0.5870
Proportion of Variance 0.274 0.190 0.145 0.136 0.0795 0.0558 0.0522 0.0345
Cumulative Proportion  0.274 0.463 0.608 0.745 0.8241 0.8799 0.9321 0.9666
```

Note that this summary presents the standard deviations for each PC and not the variances (i.e., the eigenvalues). Therefore, technically you have to square these values to obtain the eigenvalues. You can do this by calling the values from the env.pca object (recall you can type names(env.pca) to see all available results) by typing:

```
env.eig<-env.pca$sdev^2
env.eig
```

```
[1] 2.73922454 1.89544374 1.44731905 1.36396653 0.79503350 0.55833586
[7] 0.52197264 0.34456540 0.32123262 0.01290611
```

Alternatively you can use function pca.eigenval() which is contained in the BIOSTATS file (remember to source this at the beginning of your session!). Try typing (first 5 PCs are presented for clarity):

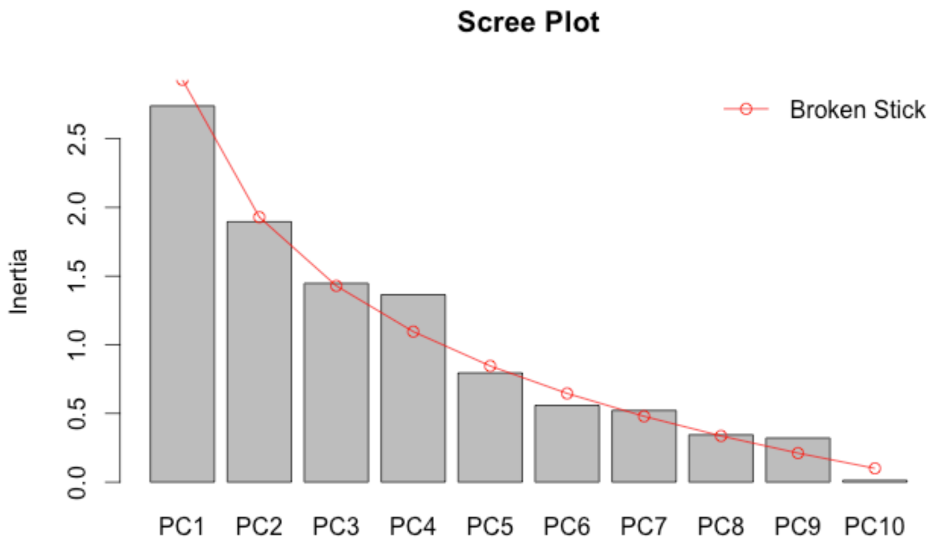
```
pca.eigenval(env.pca)
```

```
Importance of components:
                             PC1       PC2       PC3       PC4       PC5
Variance(eigenvalue)   2.7392245 1.8954437 1.4473191 1.3639665 0.7950335
Proportion of Variance 0.2739225 0.1895444 0.1447319 0.1363967 0.0795033
Cumulative Proportion  0.2739225 0.4634668 0.6081987 0.7445954 0.8240987
Broken-stick value     2.9289683 1.9289683 1.4289683 1.0956349 0.8456349
```

Notice that the values that you calculated in env.eig are identical to the summary table above.

Next, we can create a screeplot where each eigenvalue for each successive PC is depicted. If no correlation between variables existed in the original data set, the resulting PCs would be non-significant. That is, if the original variables are orthogonal, then it is impossible to collapse the variables into a lower dimensional space and the resulting PCs would be spurious. To determine whether our resulting eigenvalues are higher than expected purely by chance, we can overlay a null distribution of eigenvalues derived from the broken-stick model (these values are also presented in the summary table above); depicted by the line. In the plot below "Inertia" refers to the eigenvalue; depicted by the bars. Eigenvalues greater than the broken-stick expectation are considered to explain a statistically significance proportion of the variance in the original dataset.

```
screeplot(env.pca, bstick=TRUE)
```



From the above table and figure we can see that a large proportion of the variance in the dataset is explained by the first two PCs (46%). However, both PCs are below the null broken stick values, indicating that the variance explained is not a significant proportion of the total variance. However, ecological significance is very different from statistical significance, and I would argue that the PCA (and hence the ordination plot) is doing a good job by representing close to half of the total variation in the original data set.

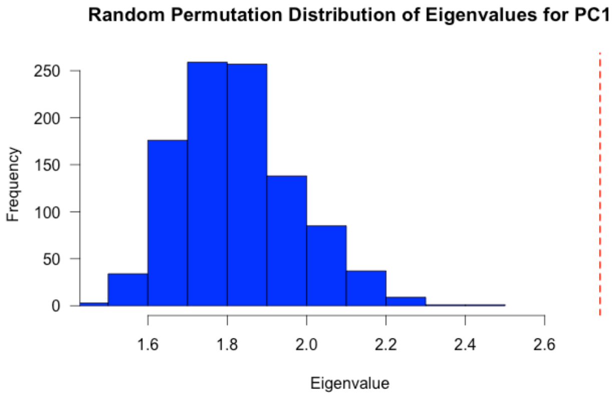
Alternatively, we may wish to test the statistical significance of the first several eigenvalues using a Monte Carlo (randomization) test. Below we conduct this analysis for the first 5 PCs (i.e., dimensions) to reduce computing time, by typing:

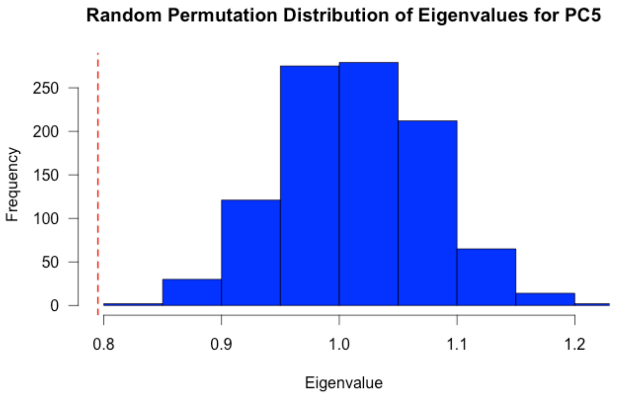
```
ordi.monte(envdata,ord='pca',dim=5)
```

This test conducts, by default, 1000 random permutations of the data matrix, permuting each column independently to remove any real correlation structure among variables, and compares the original eigenvalue to the distribution of eigenvalues under the null hypothesis of no real correlation structure. Eigenvalues with p-values less than some critical value (e.g., 0.05) are considered "significant." The random distributions of eigenvalues for each PC will be plotted, in addition to a summary table. Below are the plots for the first and last PCs.

```
Randomization Test of Eigenvalues:
              PC1    PC2    PC3    PC4    PC5
```

```
Eigenvalue 2.739 1.895 1.447 1.364 0.795
P-value    0.000 0.000 0.084 0.000 1.000
```





Eigenvectors (variable loadings):

We can examine the eigenvectors (i.e., variable loadings) on each PC (only PC1-5 are presented), by typing:

```
env.pca$rotation
```

```
                  PC1        PC2         PC3         PC4         PC5
Sinuosity  0.00429546 -0.4735237 -0.22940933  0.27361829 -0.47648306
Slope      0.26319558  0.1303184 -0.34968154 -0.46323791 -0.25803639
WDRatio    0.37836949 -0.1633701  0.31612993  0.23288139  0.30258611
SubEmbed  -0.32966590  0.2478083  0.33469255  0.25747281 -0.39866023
HabQual    0.35595505 -0.3516462 -0.24667911 -0.05692156 -0.31449155
Elev       0.23866884 -0.2113364  0.35123065 -0.53998940  0.19735782
RoadDen   -0.26905123 -0.2427914 -0.49904214  0.15497594  0.52202538
Agricult  -0.41161178 -0.2821745  0.25449588 -0.38762312 -0.18790521
HumanUse  -0.48328053 -0.3592271 -0.02179364 -0.23257076  0.10093767
BasinAre   0.14080781 -0.4845218  0.33858118  0.25851631  0.00551955
```

The loadings indicate the correlation of the original variables with each PC. Strong correlations (either positive or negative) indicate a high contribution towards the linear combination comprising each PC.

Alternatively, we can use the pca.eigenvec() function as follows:

```
pca.eigenvec(env.pca,dim=5,digits=3,cutoff=.1)
```

This function suppresses small values below the specified cutoff value (the default is 0, so all coefficients are printed) to emphasize the more important ones. Recall that eigenvector coefficients derived from the correlation matrix are proportional to the correlations between the original variables and the ordination axes, but they are NOT correlation coefficients. Nevertheless, large positive values indicate that samples positioned on the positive end of the ordination axis contain larger values of the corresponding variable, and vice versa for negative values. The dim=3 means that only the first three components are listed.

It may be more useful and interpretable to convert the eigenvector coefficients into simple correlation coefficients (i.e., Pearson product-moment correlations), as follows:

```
pca.structure(env.pca,envdata,dim=5,cutoff=.4)
```

These structure coefficients (also referred to as structure correlations, correlation loadings, or just loadings) are actual simple linear correlations between the original variables and the principal component scores. The squared structure coefficients give the percentage of variance in each original variable accounted for by each principal component.

The resulting output would look like:

```
Structure Correlations:
             PC1    PC2    PC3    PC4    PC5
Sinuosity        -0.652               -0.425
Slope      0.436        -0.421 -0.541
WDRatio    0.626
SubEmbed  -0.546         0.403
HabQual    0.589 -0.484
Elev                     0.423 -0.631
RoadDen   -0.445          -0.6         0.465
Agricult  -0.681               -0.453
HumanUse    -0.8 -0.495
BasinAre         -0.667  0.407
```

Based on the eigenvectors or the structure coefficients we generate an ecological interpretation of each principal component.

Sample scores and ordination plots:

Each sample has a score or location on each principal component axis. To see these sample scores, which are stored in the component named ‘x’ in the list object output from the prcomp() function, simply type:

```
env.scores<-env.pca$x
env.scores
```

```
            PC1         PC2          PC3         PC4          PC5         PC6
S1  -1.89423579  0.44448692  0.529427773  0.24783004  0.486832947 -0.06903318
S2  -2.45393510  2.51644089  0.514761313  0.98848294 -0.440977886  0.09479069
S3  -2.05646139 -0.70986569 -0.172173175 -1.06576107  0.117445919 -0.37566850
S4   0.80053906  1.50030939 -0.648110305 -0.06331708 -0.843362669  0.36315213
S5   0.44939256  0.75494057 -0.746555367  0.31624254 -0.982817687  0.91857748
S6   0.33102775 -2.04471858  0.786058382  0.93457905 -0.340067913  1.41184369
```

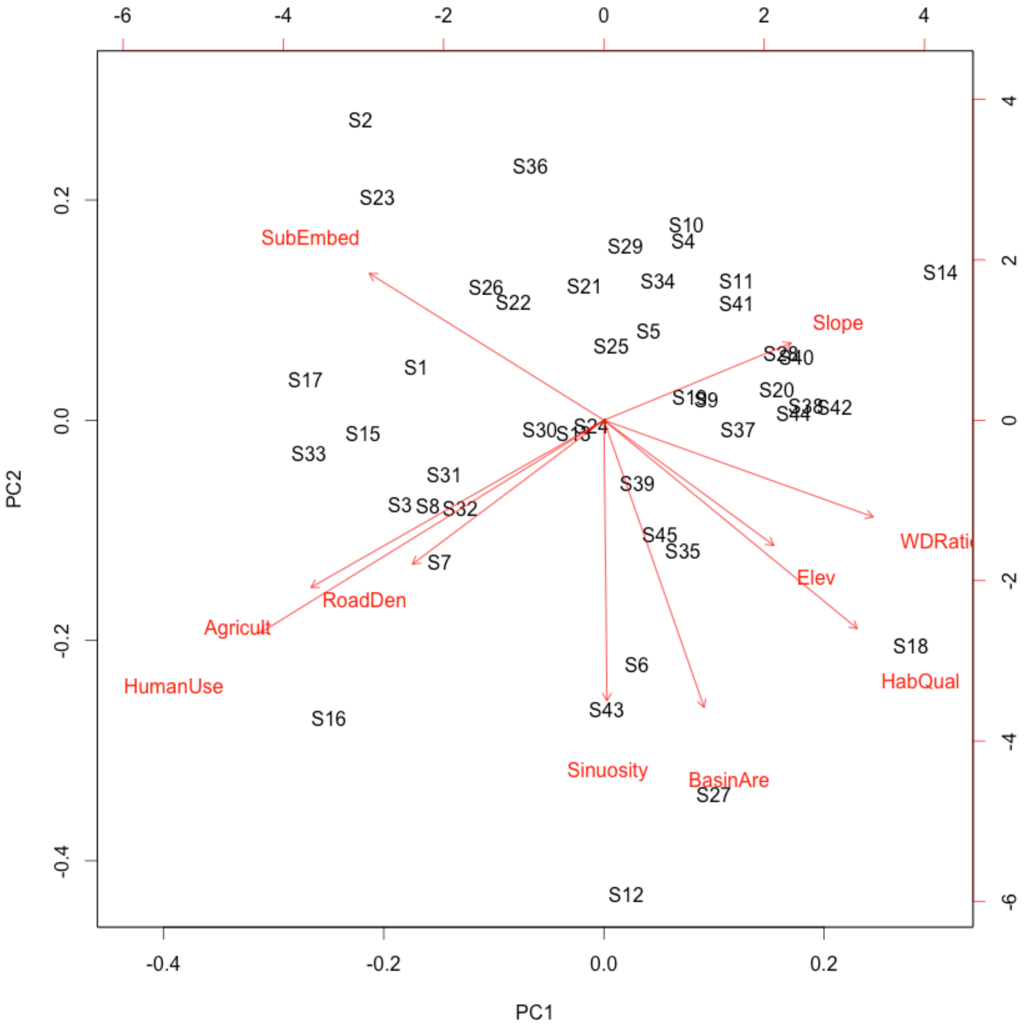
These scores are the standardized scores for each sample on each principal component axis; that is, they represent the position of each sample on each standardized principal component axis. Recall from lecture that these scores can be obtained directly by multiplying the eigenvector coefficients by the corresponding values of the standardized data for each variable and summing.

Next, we can visualize the ordination using biplot() or ordiplot().

First, let’s take the easy approach and type:

```
biplot(env.pca)
```

You will produce the following ordination biplot.



In the above figure, the vector arrows indicate the direction of variable gradients in ordination space. The longer the arrow the stronger, or more important, the respective variable is for describing the PCs. In this example, sites in the lower left corner (third quadrant) of the biplot have high values for the variables HumanUse, Agricult and RoadDen and lower values for variables with vector arrows that point towards the upper right corner (arrows that are at an increasingly oblique angle to the third quadrant). Sites located in close proximity in ordination space (e.g., S7, S32 and S31) share similar environmental features (in this case, they are all characterized by high percentages of upstream watersheds in agricultural or human land use and high road densities), and sites located far apart (e.g., S12 and S2) are very different environmentally.

Additional visualization options

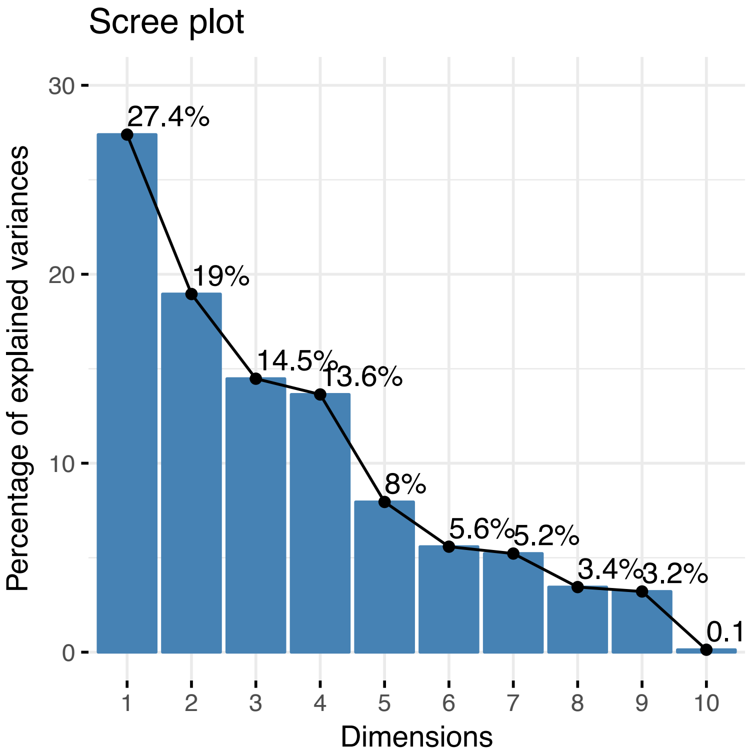
Let's use the FactoMineR and factoextra packages to produce some nicer looking outputs from PCA. In a nutshell, factoextra is an **R** package to easily extract and visualize the output of multivariate data analyses conducted in FactoMineR.

First type the following to conduct the PCA:

```
env2.pca<-PCA(envdata,graph=F)
```

Type the following to produce the scree-plot of eigenvalues:

```
fviz_screeplot(env2.pca, addlabels = TRUE, ylim = c(0, 30))
```

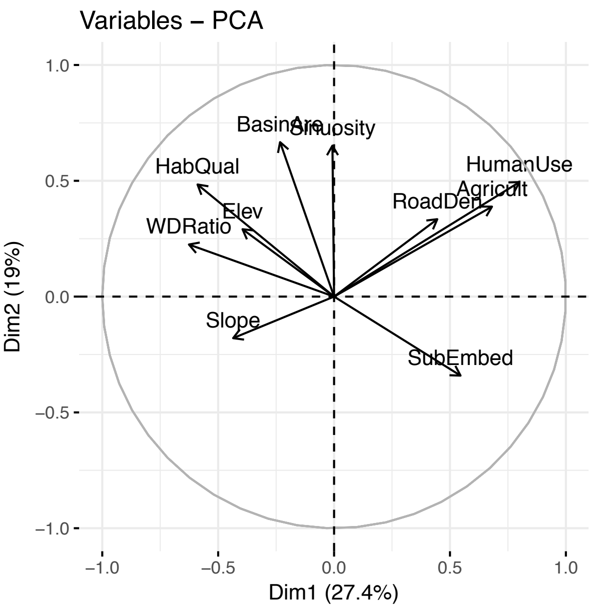


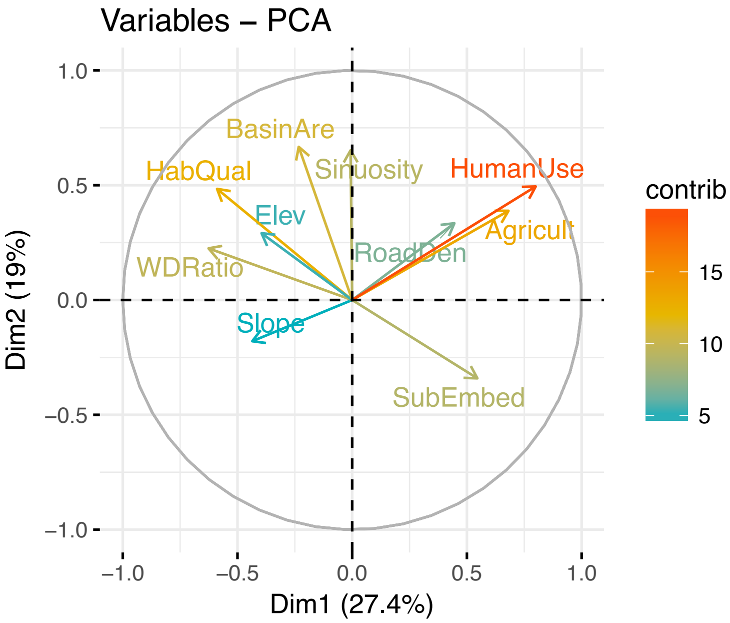
Now, let’s look at the variable loadings (i.e., eigenvectors) by typing:

```
fviz_pca_var(env2.pca, col.var = "black")
```

and a fancy looking one where total contributions of the variables are color coded:

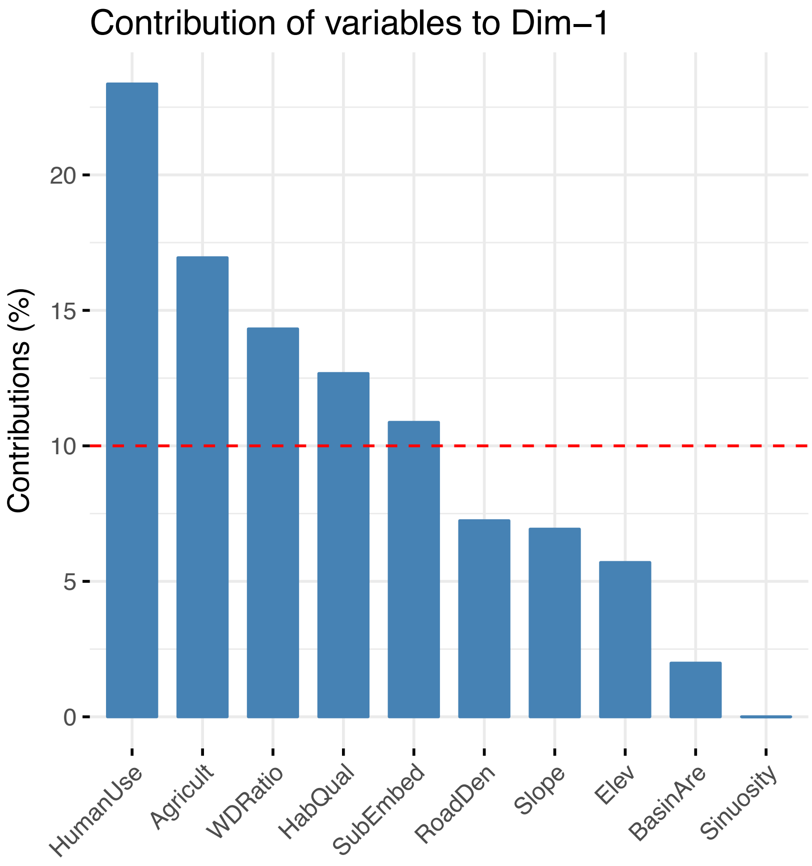
```
fviz_pca_var(env2.pca, col.var="contrib",gradient.cols = c("#00AFBB",
"#E7B800", "#FC4E07"), repel = TRUE)
```





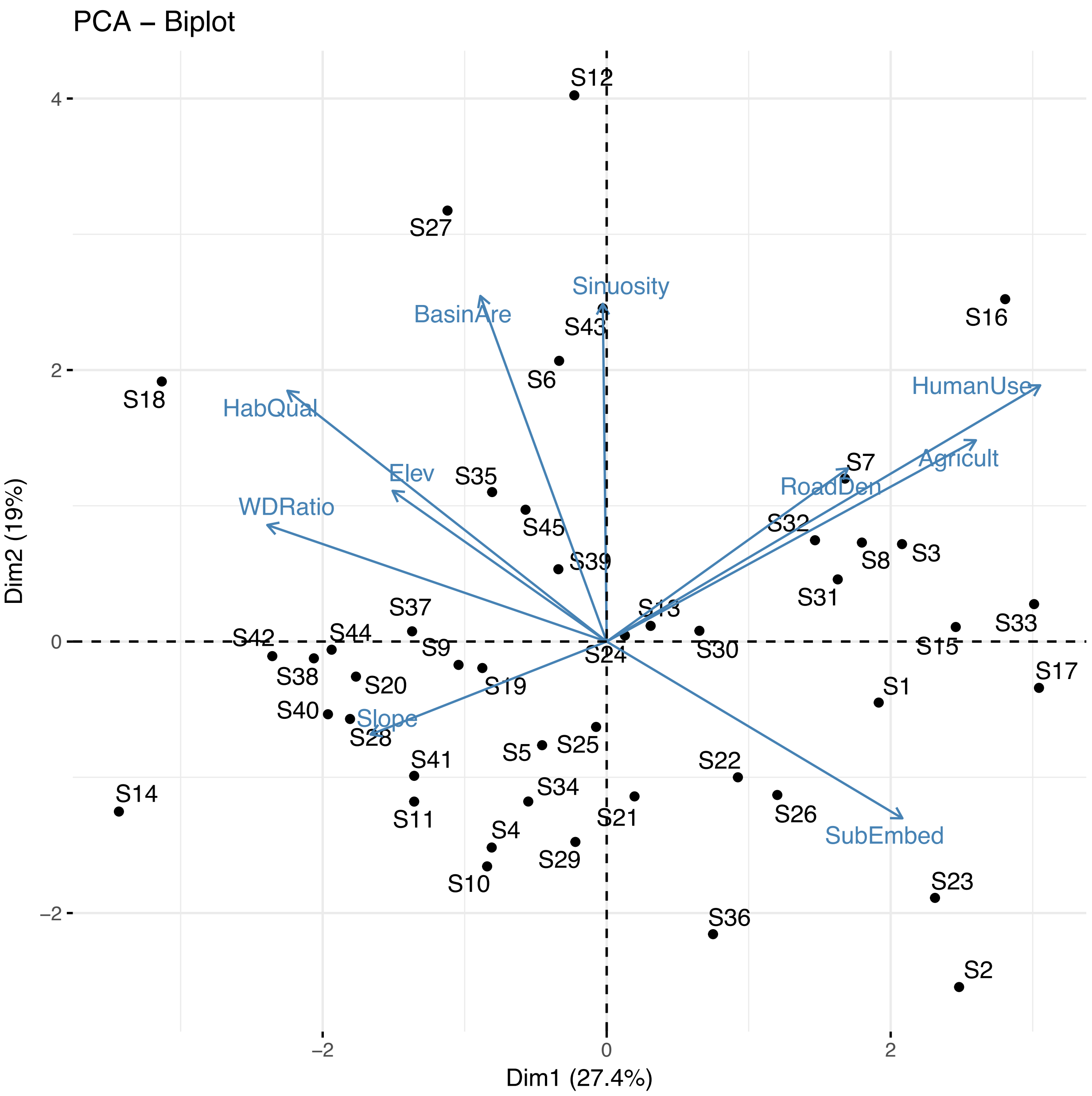
Now let's look at the relative contributions of the variables to the first PC, by typing:

```
fviz_contrib(env2.pca, choice = "var", axes = 1, top = 10)
```



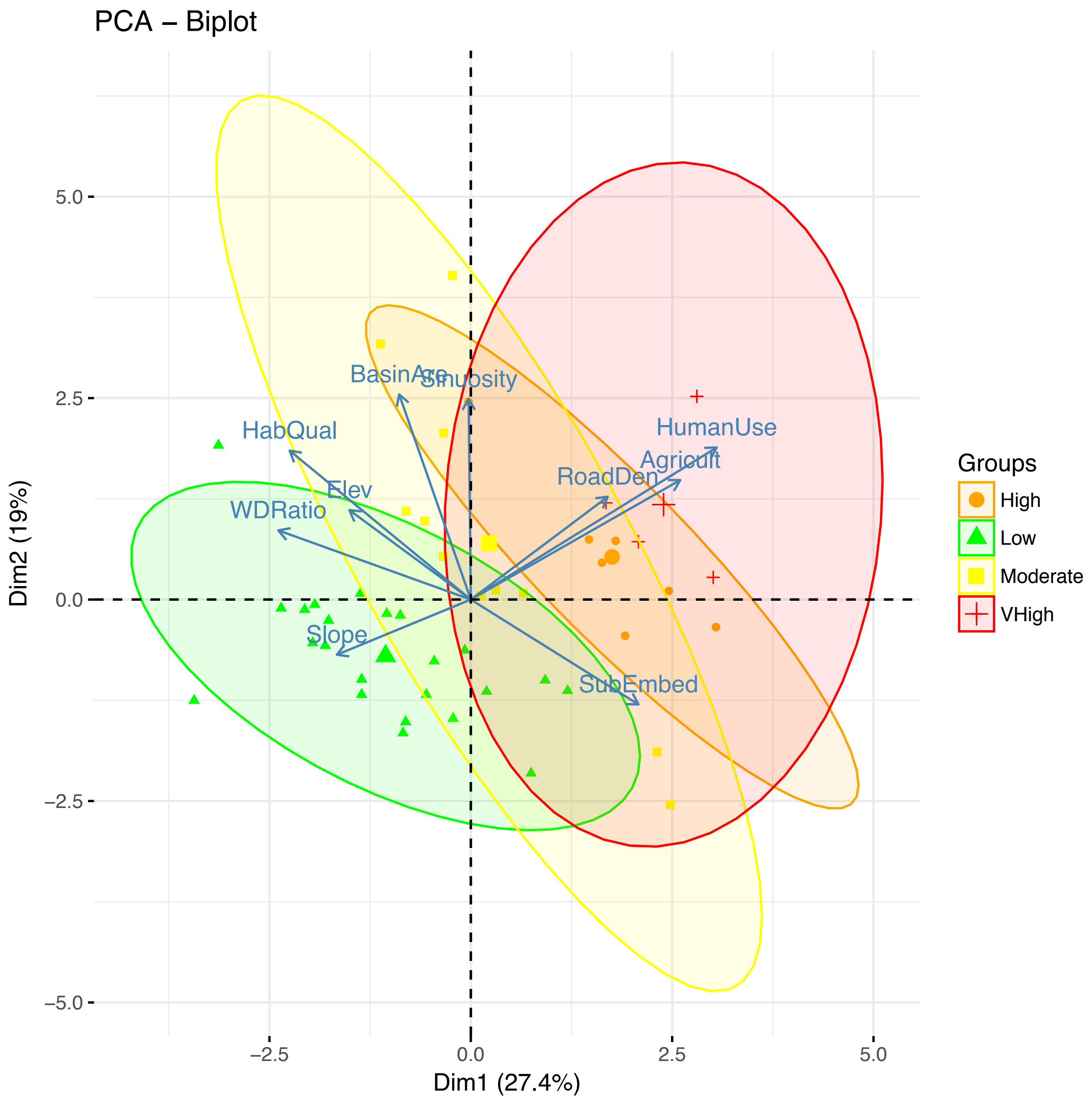
Producethe PCA biplot by typing:

```
fviz_pca_biplot(env2.pca, repel = TRUE)
```



Finally, let's now interpret the ordination plot by symbolizing each site (point) according to their level of human disturbance contained in the sitegroup dataset. We will also draw concentration ellipses. Try typing:

```
fviz_pca(env2.pca,label = "var", habillage = sitegroup$DISTURB,palette =
c("orange", "green", "yellow","red"), addEllipses = TRUE)
```



Here, the larger symbols for each disturbance category represents the centroid. You can remove the eigenvectors by using the following code:

```
fviz_pca_ind(env2.pca,habillage = sitegroup$DISTURB,palette = c("orange",
"green", "yellow","red"), addEllipses = TRUE)
```

Also, I should you mention that you can use ordiplot () in the vegan library to plot the results. Its usage is:

ordiplot(ord, choices = c(1, 2), type="points", display, xlim, ylim, ...)

Where

- **ord** is the result from an ordination
- **choices** are the axes shown
- **type** is the type of graph which may be "points", "text" or "none" for any ordination method.
- **display** only "sites" or "species". The default for most methods is to display both.
- **xlim, ylim** the x and y limits (min,max) of the plot.
- **labels** is optional text used for labels. Row names will be used if this is missing.

For additional documentation of all the arguments type `?ordiplot` in **R**.

If you want, you can try typing:

```
ordiplot(env.pca, choices = c(1, 2), type="text", display=″sites″, xlab=″PC 1
(27%)″, ylab=″PC 2 (18%)″)
```

Next, we can add the eigenvectors (i.e., the variable loadings) on the ordination, using the following command:

```
arrows(0,0,env.pca$rotation[,1]*5,env.pca$rotation[,2]*5,col=′purple′)

text(env.pca$rotation[,1]*5.2,env.pca$rotation[,2]*5.2,row.names(env.pca$rota
tion))
```

Note that we multiplied the eigenvectors by a constant (in this case, 5) so that the vectors would not be too small to interpret. Then we added the text labels to the end of the arrows.

OPTIONAL READINGS (* recommended)

Rose, C. and Smith, M. D. 1996. The Multivariate Normal Distribution. Mathematica J. 6: 32-37.

Gauch, H.G., and R.H. Whittaker. 1972. Comparison of ordination techniques. Ecology 53: 868-875.

Goodall, D.W. 1954. Objective methods for the classification of vegetation. III. An essay in the use of factor analysis. Australian Journal of Botany 2: 304-324.

James, F.C. and McCulloch. 1990. Multivariate analysis in ecology and systematics: Panacea or Pandora's box. Annual Review in Ecology and Systematics 21:129-166.

Pearson, K. 1901. On lines and planes of closest fit to systems of points in space. Philosophical Magazine 2:559-572.

EXERCISE

Purpose

Upon completion of this chapter, you should be able to do the following: (1) Carry out a principal components analysis; (2) assess how many principal components should be considered in an analysis; (3) interpret principal component scores; (4) determine when a principal component analysis may be based on the variance-covariance matrix, and when the correlation matrix should be used; and (5) consider how principal component scores may be used in further analyses.

Tasks

- Perform a principal component analysis using the covariance matrix.
    - How many PCs are required to explain 90% of the total variation for this data? Are these statistical significant? How did you test for this?
    - For the number of components in part a, give a brief interpretation of each component.
    - Interpret the bi-plot (i.e., ordination with object scores and eigenvectors)
- Perform a principal component analysis using the correlation matrix. Repeat questions 1a-c.
- Conduct a PCA on the species occurrence dataset. What are your results and was this a good decision?