ORDINATION
MAKING SENSE OF COMPLEX DATASETS

Alejandro Ordonez

Assistant Professor - Department of Bioscience

Section for Ecoinformatics & Biodiversity

Center for Biodiversity Dynamics in a Changing World (BIOCHANGE)

ORDINATION
WHAT WE WILL TALK ABOUT TODAY

- What is Ordination
- Ordination as a multivariate approach
- Correlations based multivariate approaches
  - Principal Components Analysis (PCA)
  - Correspondence Analysis (CA)
- Distance based multivariate approaches
  - Principal Coordinates Analysis (PCoA)
  - (Non) metric dimensional Scaling (N)MDA

Any questions?

Ready to start?

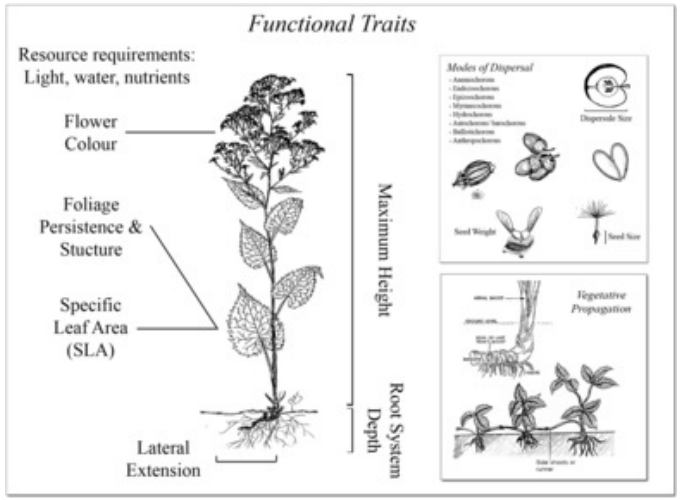
THE PROBLEM AT HAND

Biological/ecological datasets are multidimensional

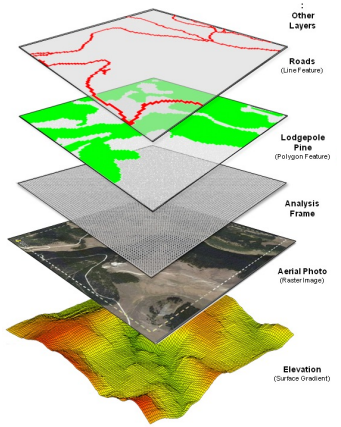
What is multidimensional? → multiple descriptors are recorded from a number of objects (i.e., replicate sampling or experimental units)

- **Organisms**: morphological or physiological measurements
- **Ecological sampling units**: the variables might be physicochemical measurements or species abundances

Organism attributes



Ecological sampling units



MULTIVARIATE DATA DESCRIPTION
ORDINATION VS CLASSIFICATION

While **CLASSIFICATION** looks for discontinuities in a dataset...

ORDINATION extracts the main trends in the form of continuous axes

The aim of ordination methods:

Represent the data along a reduced number of **orthogonal axes**, representing, in decreasing order, the main trends of variation of the data.

ORDINATION - WHAT IS DONE?

Ordination is about ***re-arranging*** multivariate objects by assigning each object new coordinates such that similar points are near each other based on their descriptors.

Methods include:

- PCA
- PCoA

Euclidean variable reduction

Think a data set whose variables are normally distributed

- CA
- (D)CO
- (N)MDS

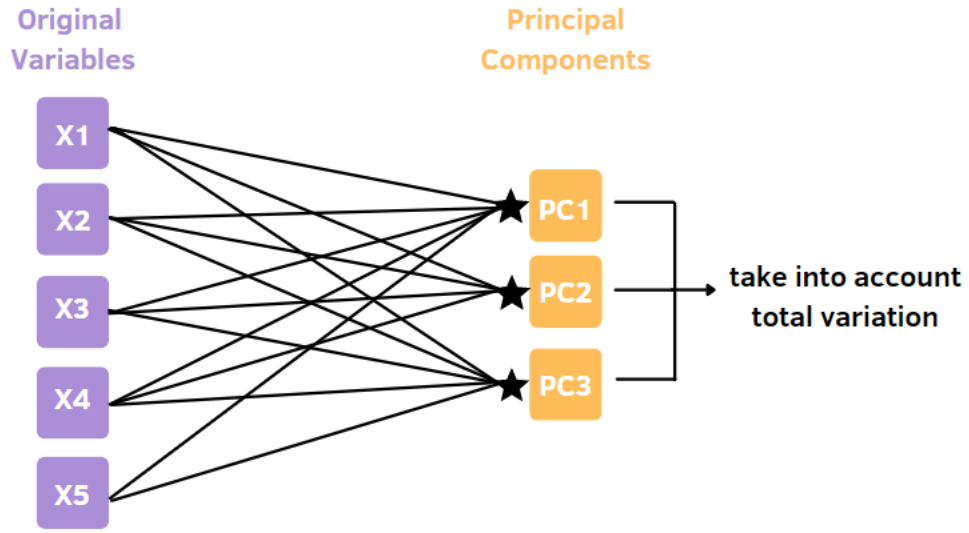
Rank order variable reduction

Think a data set whose variables are **NOT** normally distributed

ORDINATION
WHY GO TO ALL THE TROUBLE?

Variable reduction:

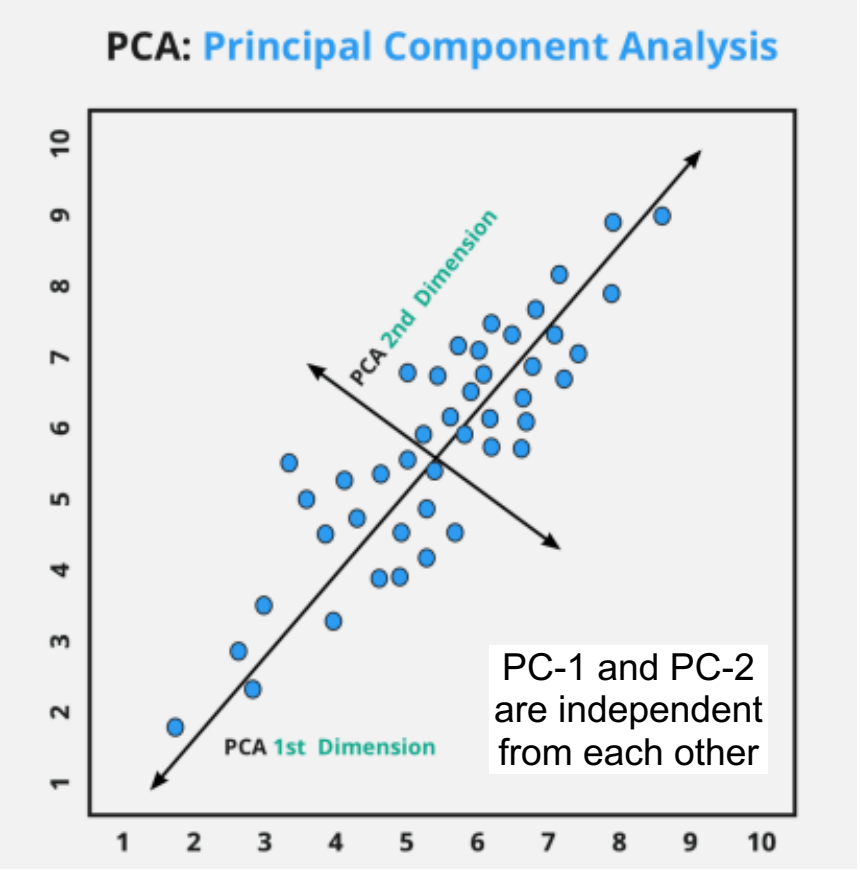
- Reduce the number of dimensions to a more manageable number while...
  - Minimising (in some way) distortion in the data.
  - Maximising (in some way) the resulting information.



ORDINATION
WHY GO TO ALL THE TROUBLE?

Solve problems of collinearity of the data:

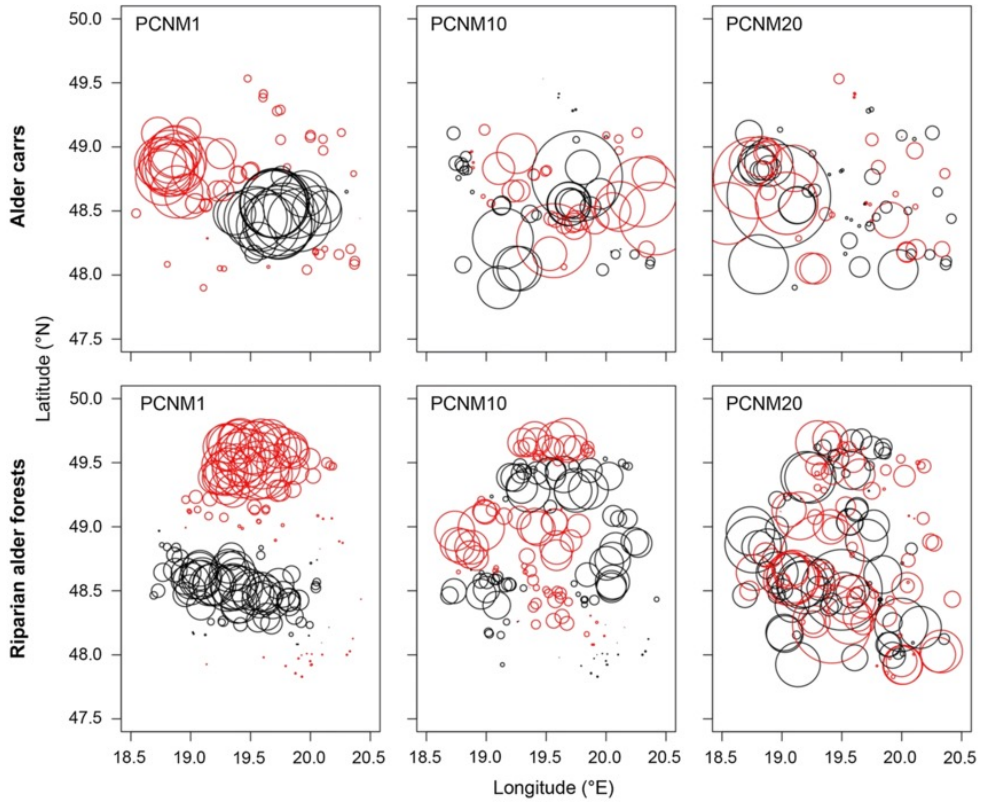
- New axes along which descriptors/objects are arranged are independent of each other...
  - can be used as predictors in a regression.



ORDINATION
WHY GO TO ALL THE TROUBLE?

Solve Autocorrelation problems:

- You can develop a series of axes of variation that describe the "similarity" between observations → can be used as predictors in a regression.



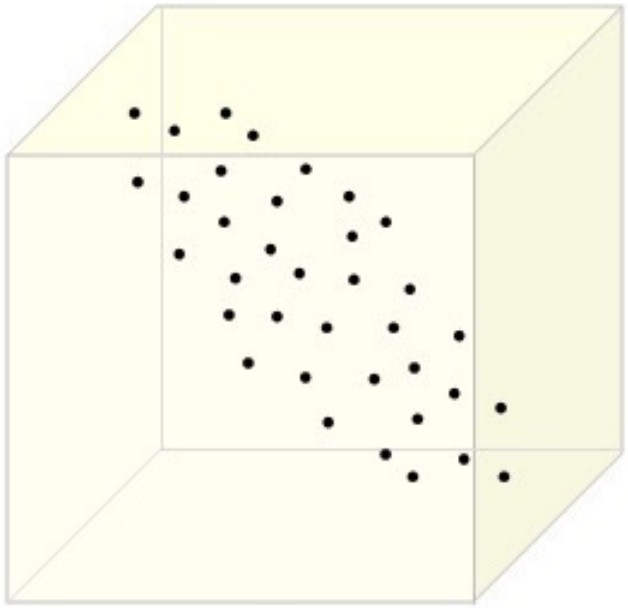
So far so good?

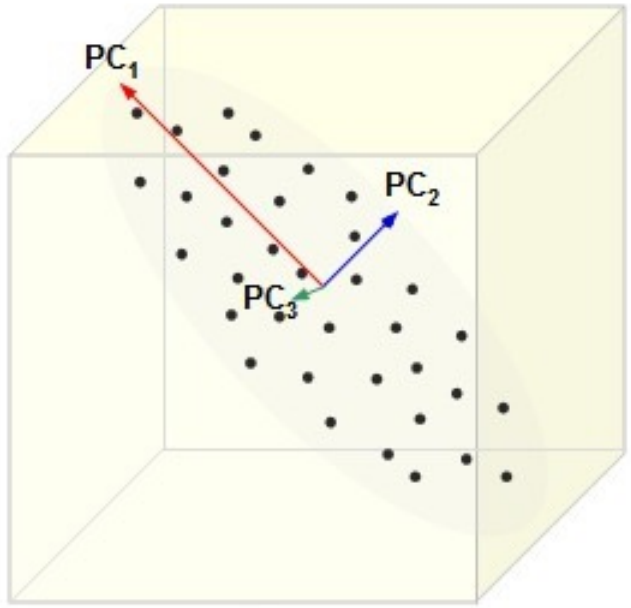
Any questions?

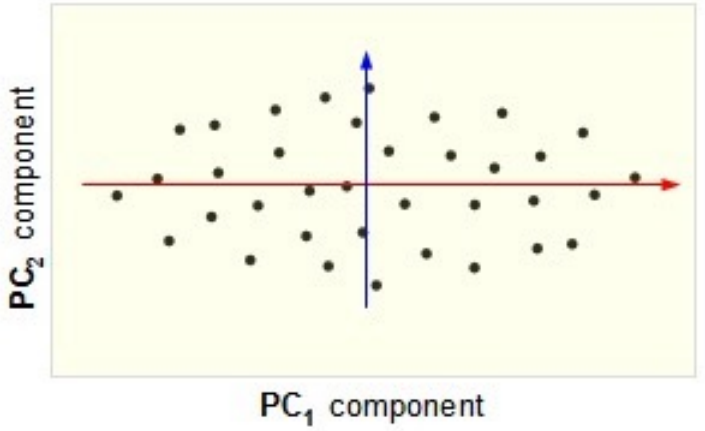
Ready to move on?

ORDINATION:
HOW DOES LOOK LIKE GRAPHICALLY?

Original data	Dimension reduction	Ordination





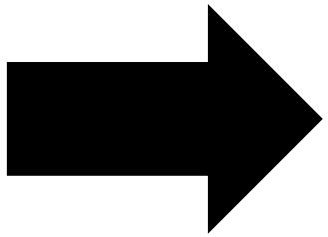


http://cnx.org/content/m11461/latest/

ORDINATION:
HOW DOES LOOK LIKE NUMERICALLY?

Original coordinates

	Var 1	Var 2	Var 3	Var 4	Var 5
Obs 1	0.471	1.07	-1.60	1.145	-0.724
Obs 2	1.37	0.900	1.03	-0.696	0.793
Obs 3	-1.27	-0.764	0.939	-1.03	-1.49
Obs 4	0.527	0.967	0.778	1.05	0.821
Obs 5	-0.911	-0.618	0.053	-0.936	-1.13
Obs 6	0.030	-0.098	0.562	-0.702	0.759
Obs 7	0.914	0.286	-0.904	1.27	-0.059
Obs 8	-1.14	-1.74	-0.855	-0.102	1.03



Ordinated coordinates

	PCA 1	PCA 2	PCA 3	PCA 4	PCA 5
Obs 1	-1.540	1.621	-0.789	0.259	-0.320
Obs 2	-0.968	-1.902	-0.079	0.570	0.080
Obs 3	2.246	-0.187	-1.078	-0.365	0.112
Obs 4	-1.453	-0.770	0.184	-0.919	-0.129
Obs 5	1.656	0.303	-0.711	0.207	-0.039
Obs 6	0.295	-0.971	0.544	0.213	-0.164
Obs 7	-1.488	0.965	0.000	0.005	0.481
Obs 8	1.252	0.940	1.928	0.030	-0.021

How can we do this transformation?

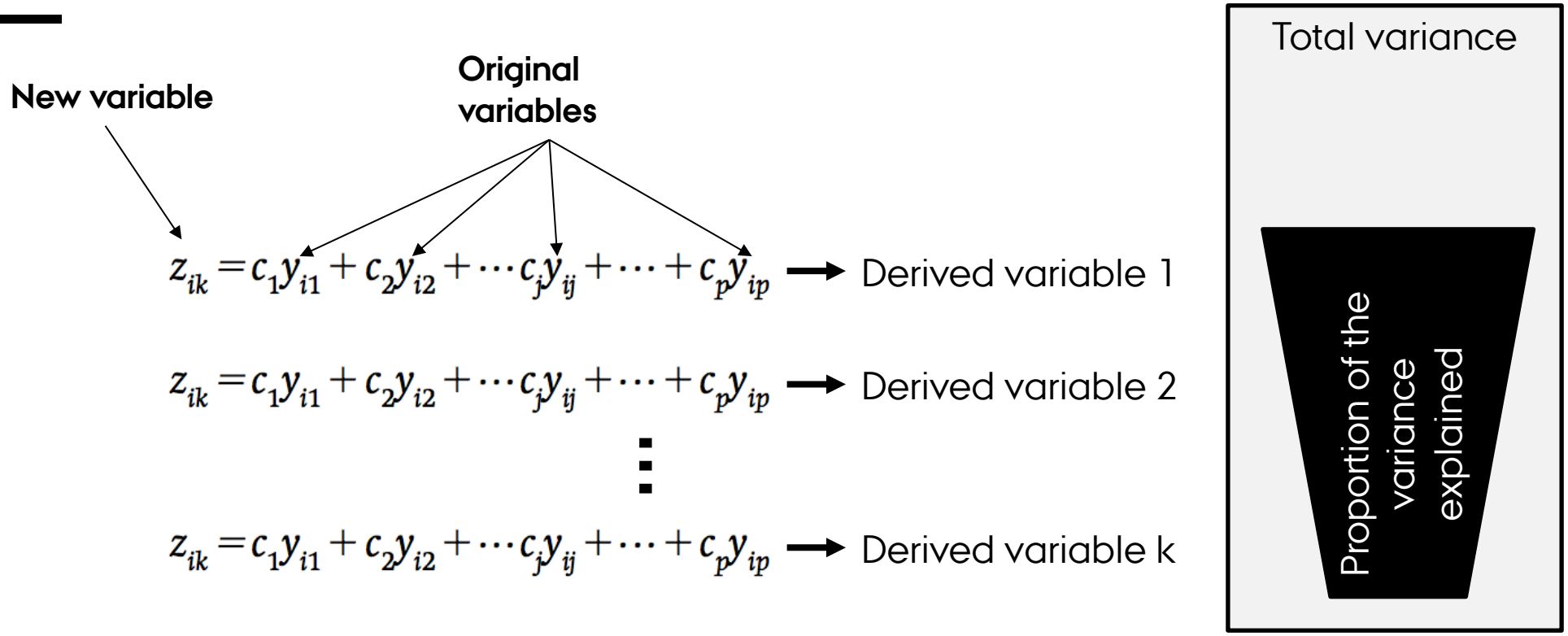
ORDINATION BASICS:
EUCLIDEAN VARIABLE TRANSFORMATION

The goal of a Euclidean Variable Transformation is deriving a **linear combinations of the variables** that summarise the variation in the original data set.

What does this mean?:

- We are "**consolidating**" the variance from a data matrix into a **new set of variables**.
- The new variables are a linear combination of the original variables.
- The new variables are independent/uncorrelated. → **orthogonal.**

ORDINATION BASICS :
EUCLIDEAN VARIABLE TRANSFORMATION



Depending on the analysis z_{ik} can be called: **discriminant functions, canonical functions or variates, principal components or factors.**

ORDINATION BASICS:
EIGENVECTORS AND EIGENVALUES

Eigenvalues (λ):

- The amount of the original variance explained by each of the new derived variables.
- Eigenvalues are population parameters.
  - They change based on the sample → a new or a larger sample changes these!

Eigenvector:

- Lists of the coefficients or weights showing how much each original variable contributes to each new derived variable (c_1 to c_p).
- The c_p coefficients can be rescaled in different ways and are often represented as u_j, v_j, or w_j.

ORDINATION BASICS:
EUCLIDEAN VARIABLE TRANSFORMATION

Eigenvector Axis 1

$$z_{ik} = c_1 y_{i1} + c_2 y_{i2} + \cdots c_j y_{ij} + \cdots + c_p y_{ip}$$ → Derived variable 1

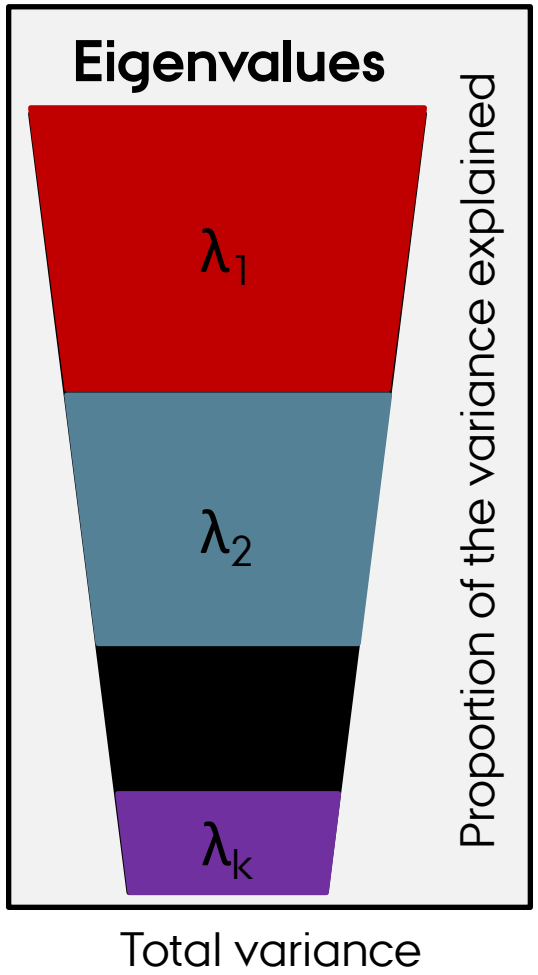
Eigenvector Axis 2

$$z_{ik} = c_1 y_{i1} + c_2 y_{i2} + \cdots c_j y_{ij} + \cdots + c_p y_{ip}$$ → Derived variable 2

⋮

Eigenvector Axis k

$$z_{ik} = c_1 y_{i1} + c_2 y_{i2} + \cdots c_j y_{ij} + \cdots + c_p y_{ip}$$ → Derived variable k



So far so good?

Any questions?

Ready to move on?

PRINCIPAL COMPONENTS ANALYSIS
PCA

One of the most commonly used multivariate statistical techniques, and it is also the basis for some others.

PCA transforms the original variables into a new set of uncorrelated variables.

- Principal components or factors.
- Object new positions along each component are called z-scores.

WHICH ARE THE PCA ASSUMPTIONS?

PCA is a linear geometric transformation that preserves **EUCLIDEAN** distances.

- Creates a series of orthogonal axes based on linear associations.
- The new positions can be plotted in a traditional cartesian system.
- It is applicable to any multivariate set of **continuous variables** that are **multivariate normal.**

WHICH ARE THE PCA ASSUMPTIONS?

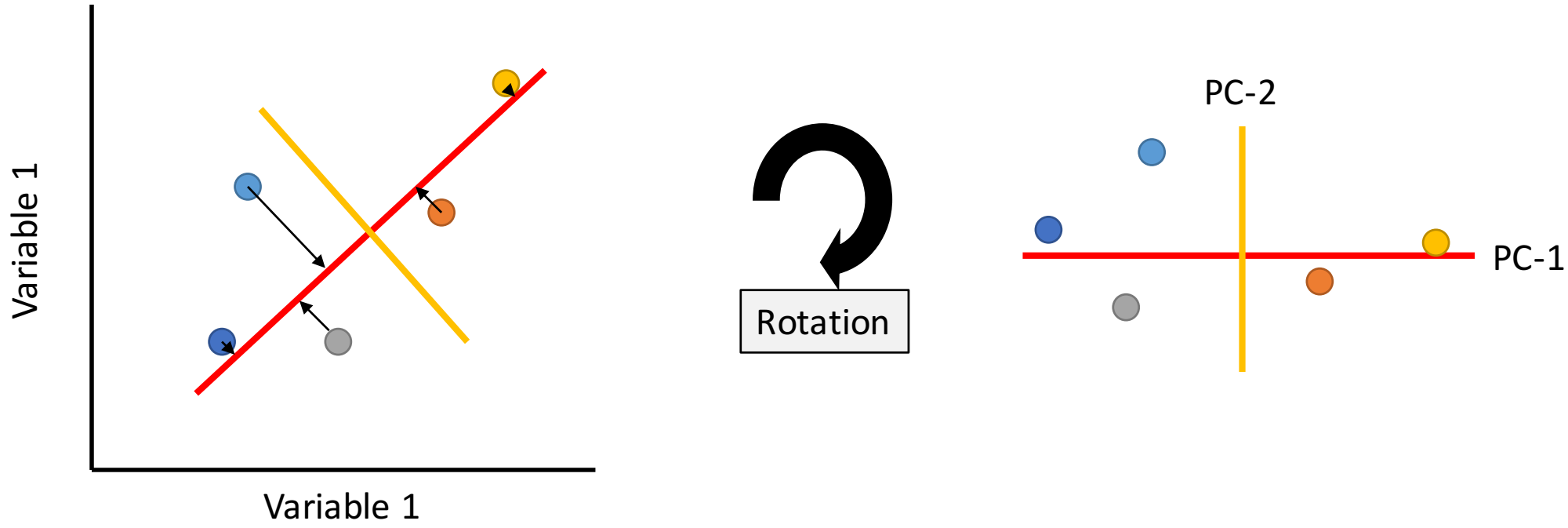
PCA are

- **Useful for complex environmental data** associated to this information.
- **Not useful for ecological community data** (abundances)
  - Why? These are often non-normal → **But see Borcard et al (2011) Numerical Ecology for a way you can use it with species data.**

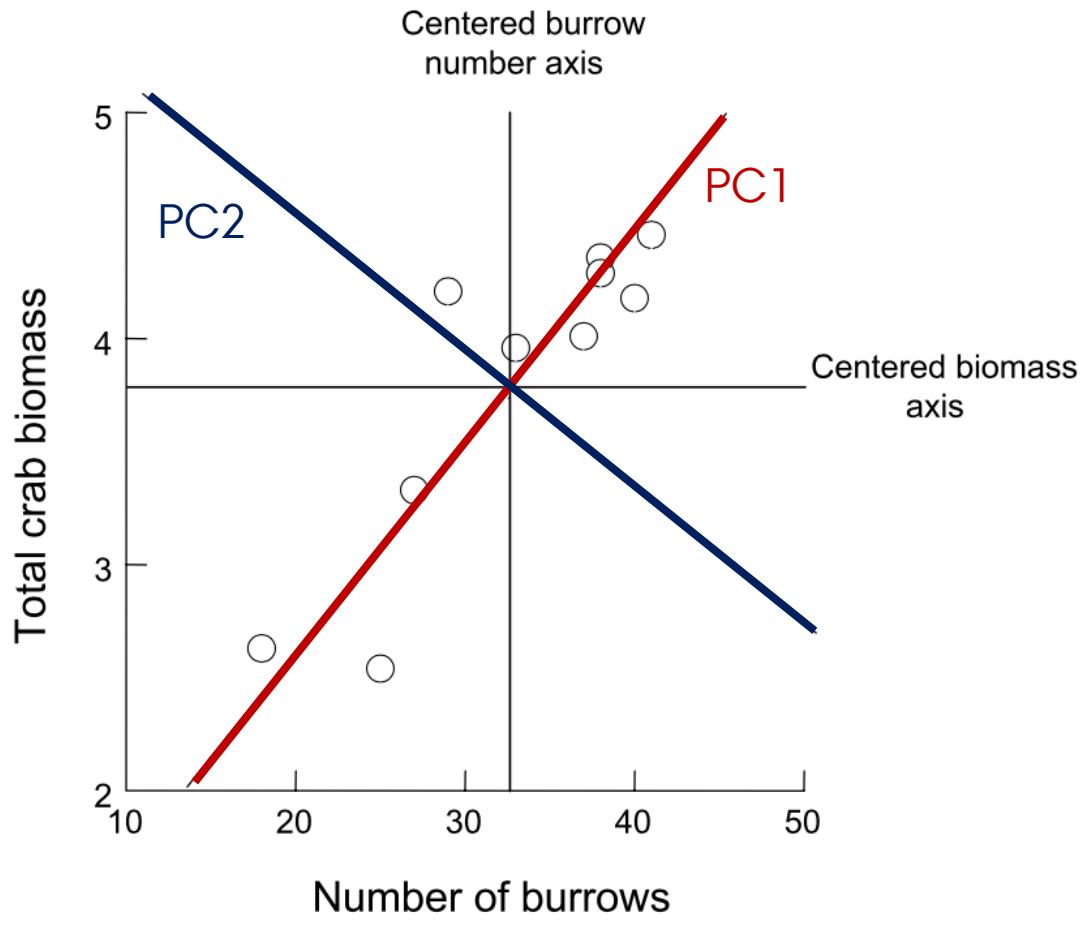
PRINCIPAL COMPONENTS ANALYSIS (PCA)

What is a PCA doing?:

- PCA can be viewed as a rotation of the principal axes of variation after centring to the mean



PRINCIPAL COMPONENTS ANALYSIS (PCA)



What do I need to make this rotation?

- The eigenvectors matrix is used as a rotational matrix

How does this rotation take place?

- Matrix multiplication of the original data and a eigenvectors matrix for the objects

PRINCIPAL COMPONENTS ANALYSIS (PCA)

A PCA by hand

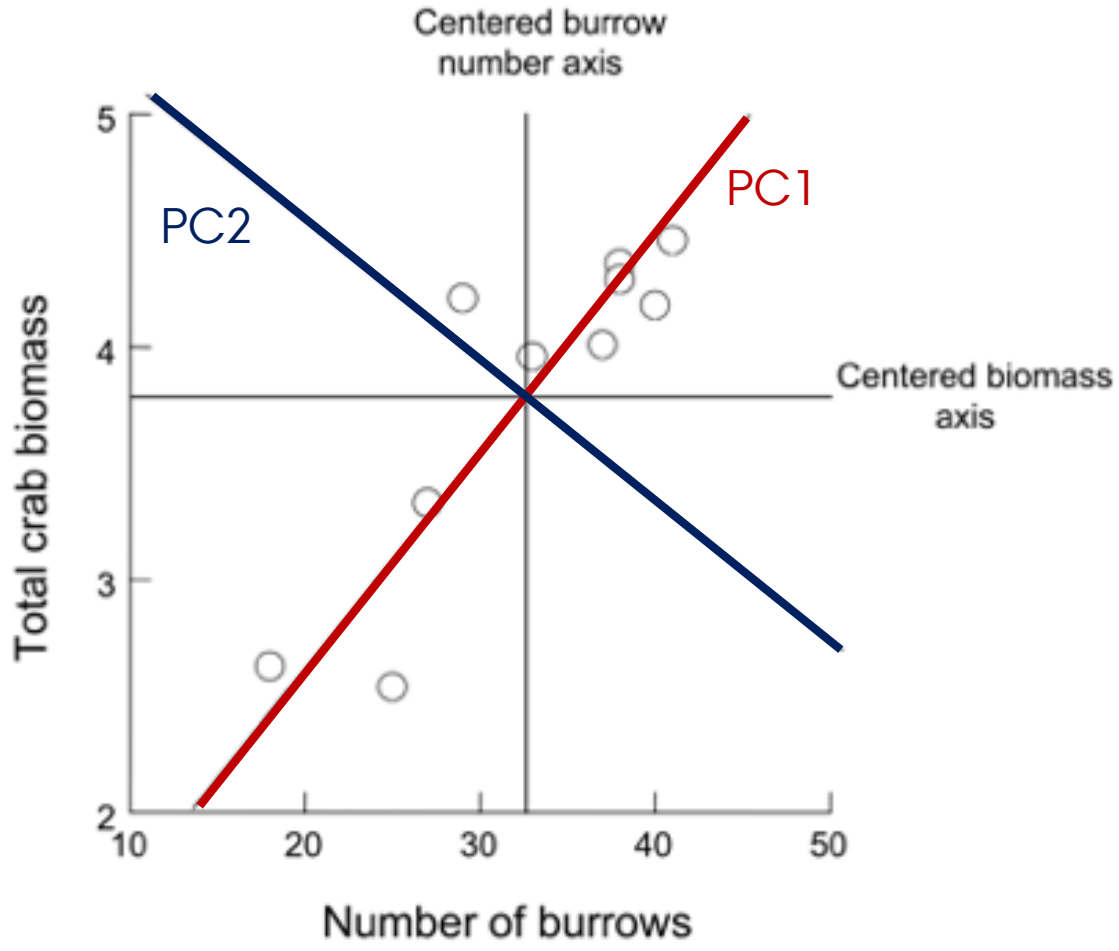
```
# The A data frame is a random example
A = matrix(runif(40),ncol = 5)
# We centre the original data
A = scale(A)
pca = eigen(cov(A))
pca
eigen() decomposition
$values
[1] 2.85711173 1.25955149 0.47876091 0.35415461 0.05042126
$vectors
           [,1]       [,2]       [,3]       [,4]       [,5]
[1,] -0.4569472  0.28275491  0.64234686 -0.54474600 -0.0434883
[2,] -0.4243838  0.52377391 -0.50877502  0.07029859 -0.5308157
[3,] -0.2979562 -0.76011387  0.06070147 -0.02724050 -0.5736046
[4,]  0.5117588  0.26016024  0.48519784  0.32376386 -0.5746125
[5,] -0.5103162  0.01594062  0.29906000  0.76989923  0.2390429
```

Note I use the Covariance matrix

Eigenvalues (λ)

Eigenvectors

PRINCIPAL COMPONENTS ANALYSIS (PCA)



A matrix multiplication of predictors and eigenvectors

A PCA by hand

```
# Estimate the PCA positions of each observation
pA = A %*% pca$vectors
pA
```

```
          [,1]    [,2]    [,3]    [,4]    [,5]
[1,]  0.2519  0.3819 -1.6000 -0.0869  0.1131
[2,]  2.8341  0.0157  0.0525 -0.0420 -0.2557
[3,] -1.7016 -0.6275  0.1870  0.1672 -0.3464
[4,]  1.7111  0.2273  0.4506 -0.5922  0.0196
[5,]  0.1843  1.2451  0.7008  0.5105  0.2946
[6,] -1.8173 -0.4385  0.2348 -1.0828  0.1304
[7,]  0.1839 -2.1569  0.0312  0.6726  0.1797
[8,] -1.6463  1.3529 -0.0569  0.4536 -0.1353
```

PRINCIPAL COMPONENTS ANALYSIS (PCA)

An automatic version of PCA in R is implemented in two methods in the `stats` package (which is a default package in R)

- `princomp()` → Spectral Decomposition Approach.
- `prcomp()` → **Singular Value Decomposition Approach.**

The method is part of other packages

- `rda()` form the `vegan` package.
- `pca()` from the `ada` package.

These are the preferred approaches!!

Why?

You don't need to think if you should use a correlation or a covariance matrix as input

PRINCIPAL COMPONENTS ANALYSIS: EIGENVECTORS AND EIGENVALUES

Singular value decomposition (SVD)

Transform an association matrix of variables using both the associations between objects and an Eigenvector matrix

$$Y = Z L^{1/2} U'$$

- Y: A matrix of centred data (*n x p*)
- L: Eigenvalue matrix (*n x p*)
- Z: Matrix of eigenvectors for variables (*n x p*)
- U: Matrix of eigenvectors of objects (*p x p*)

Type of data used defines the equivalence between methods:
raw data ➔ SDC of the Squ & Cross Prod matrix (SSCP)
centred data ➔ SDC of the covariance matrix
centred and standardized ➔ SDC of the correlation matrix

So far so good?

Any questions?

Ready to move on?

EXTRACTING INFORMATION OF A PCA

```
> A = matrix(runif(80),ncol = 5) # Generate a random data set.
> pca.1 = prcomp(A, scale=TRUE) # Determine the Principal
Components.
> pca.1
Standard deviations (1, .., p=5):
[1] 1.2952528 1.1212786 0.9895085 0.7936766 0.6752814
Rotation (n x k) = (5 x 5):
             PC1        PC2         PC3        PC4        PC5
[1,]  0.4660372 -0.3331315  0.52570083 -0.5096097 -0.3684689
[2,]  0.5079742  0.1690350 -0.62387417  0.1138295 -0.5578649
[3,] -0.3726916  0.6051757 -0.07786541 -0.6683264 -0.2052806
[4,]  0.6020957  0.3326498 -0.07300934 -0.2356989  0.6825984
[5,]  0.1528179  0.6193198  0.56835400  0.4744751 -0.2119829
```

Eigenvectors

EXTRACTING THE USEFUL INFORMATION OF A PCA

$Y_5=-(0.37*X_1)-(0.56*X_2)-(0.20*X_3)+...$

```
Standard deviations (1, .., p=5):
[1] 1.2952528 1.1212786 0.9895085 0.7936766 0.6752814
Rotation (n x k) = (5 x 5):

           PC1        PC2         PC3        PC4        PC5
[1,]  0.4660372 -0.3331315  0.52570083 -0.5096097 -0.3684689
[2,]  0.5079742  0.1690350 -0.62387417  0.1138295 -0.5578649
[3,] -0.3726916  0.6051757 -0.07786541 -0.6683264 -0.2052806
[4,]  0.6020957  0.3326498 -0.07300934 -0.2356989  0.6825984
[5,]  0.1528179  0.6193198  0.56835400  0.4744751 -0.2119829
```

$Y_1=(0.47*X_1)+(0.51*X_2)-(0.37*X_3)+...$

EXTRACTING THE USEFUL INFORMATION OF A PCA

```
A = matrix(runif(80),ncol = 5) # Generate a NEW random data set.
pca.1 = prcomp(A, scale=TRUE) # Determine the Principal
Components.
pca.1$x
```

```
              PC1          PC2         PC3          PC4         PC5
[1,] -1.1577468 -0.155679408  0.02843014  0.79537856 -0.59433350
[2,] -1.3862411  0.084352130 -1.44746588  0.89987729  0.06434587
[3,]  1.2835030 -0.109626990  1.63156616 -0.04464804 -1.04763792
[4,]  0.2852092 -1.189488968  0.43008110  1.56039662  1.28183884
[5,]  0.8246618 -0.967654874  0.07214084  0.38968351 -0.87802227
...
```

Position of each observation within the Principal Component Space

EXTRACTING THE USEFUL INFORMATION OF A PCA

```
A = matrix(runif(80),ncol = 5) # Generate a NEW random data set.
pca.1 = prcomp(A, scale=TRUE) # Determine the Principal
Components.
summary(pca.1) # Summary of the Principal Components Analysis.
Importance of components:
                          PC1    PC2    PC3    PC4    PC5
Standard deviation     1.319 1.0990 0.9685 0.7996 0.68918
Proportion of Variance 0.348 0.2416 0.1876 0.1279 0.09499
Cumulative Proportion  0.348 0.5896 0.7772 0.9050 1.00000
```

Spread within each PC-axis

Standard deviation in each direction

EXTRACTING THE USEFUL INFORMATION OF A PCA

```
A = matrix(runif(80),ncol = 5) # Generate a NEW random data set.
pca.1 = prcomp(A, scale=TRUE) # Determine the Principal
Components.
summary(pca.1) # Summary of the Principal Components Analysis.
Importance of components:
                          PC1    PC2    PC3    PC4     PC5
Standard deviation     1.1319 1.0990 0.9685 0.7996 0.68918
Proportion of Variance 0.3480 0.2416 0.1876 0.1279 0.09499
Cumulative Proportion  0.3480 0.5896 0.7772 0.9050 1.00000
```

Explained variance
Eigenvalues (λ_i)

So far so good?

Any questions?

Ready to move on?

HOW MANY PC'S?

No clear-cut rules, only rules of thumb

- **Possibility - 1**: Cumulative proportion should be at least 0.7 or 0.8 (i.e. 70/80% of variance is captured)

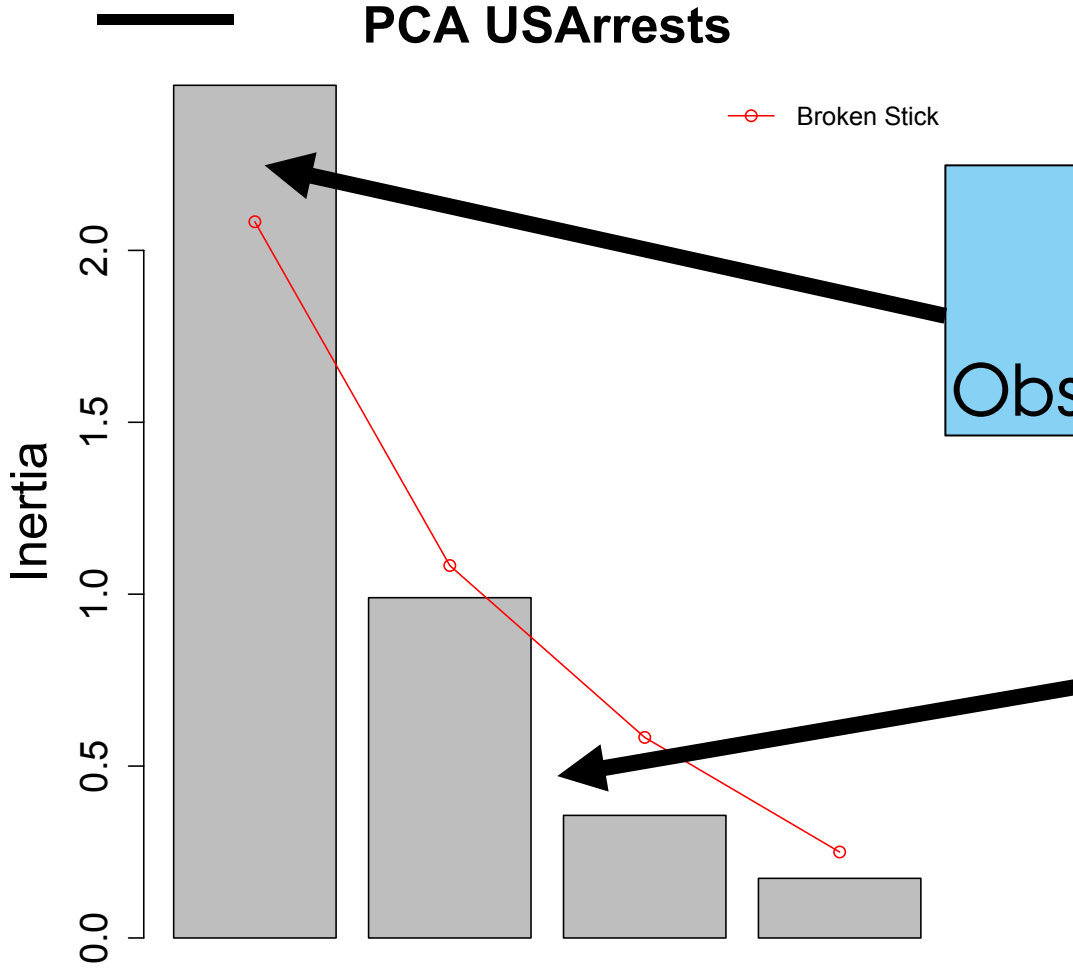
- **Possibility - 2**: Look at scree-plot and keep:
  - PC's before the "elbow" (if there is any…)
  - PC's axes whose eigenvalues are larger than the length of the corresponding piece form a theoretical broken stick

HOW MANY PC'S?
CUMULATIVE PROPORTION

With a 70-to-80% variance explain cut-off means only PC-1 and PC-2 need to be kept!

```
> data(USArrests)
> pca.2 <- prcomp(USArrests, scale = TRUE)
> summary(prcomp(USArrests, scale = TRUE))
Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.5749 0.9949 0.59713 0.41645
Proportion of Variance 0.6201 0.2474 0.08914 0.04336
Cumulative Proportion  0.6201 0.8675 0.95664 1.00000
```

VISUALISING AN PCA
SCREEPLOT



Broken stick approach

Only PC-1 should be kept

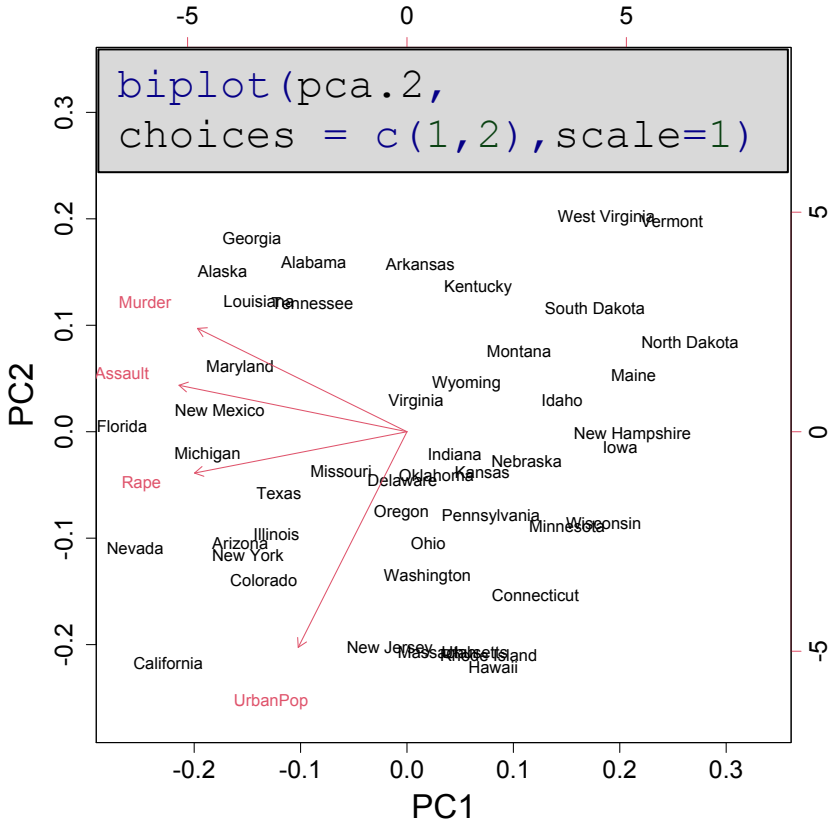
Observed var is larger than expected

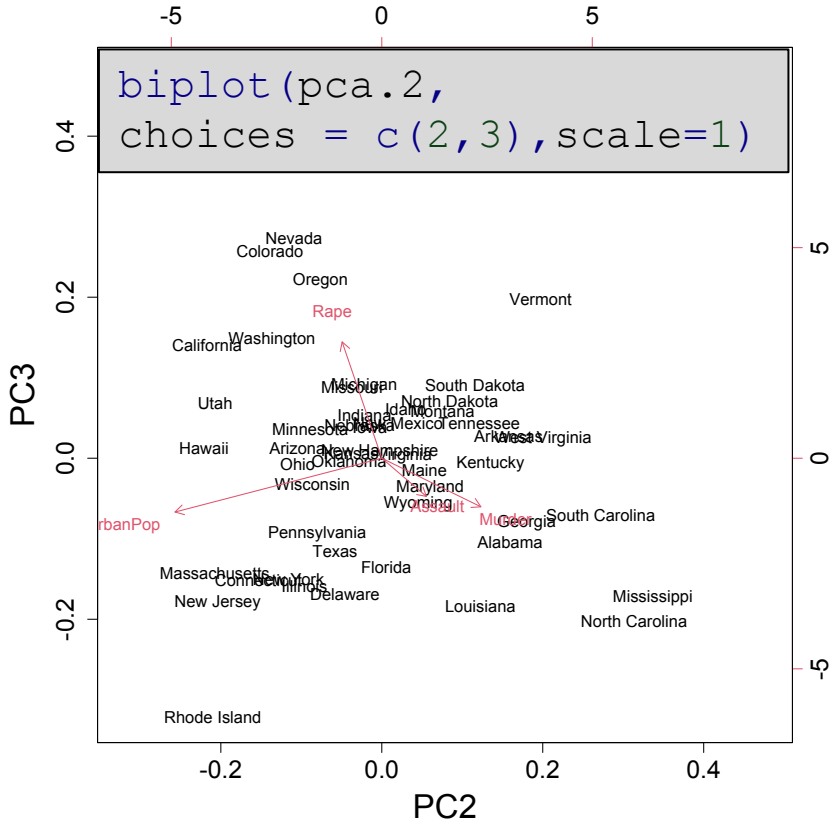
Elbow approach

PC-1 and PC-2 should be kept

VISUALISING AN PCA – USING `BIPLOT()`

PCA USArrests





How to read a PCA biplot

1) The **names in black** represent each location
2) The **red vectors** represent the direction and importance of each evaluated variable in the context of the plotted PC-axis.
3) The angles between **red vectors** reflect their correlations.
   These position are defined by the eigenvectors each PC-axis

So far so good?

Any questions?

Ready to move on?

CORRESPONDENCE ANALYSIS (CA)

What it is?

- An ordination technique the focuses on **categorical** rather than **continuous** data → *this makes it useful for Species data*
- Appropriate with unimodal (non-linear) response curves. → *This is a requirement for PCA*

The goal?

- Exploring the relationship between variables in a contingency table.
- Define a series of orthogonal axes and, for each item in a table, a set of scores.
- The position along these axes preserves the $\chi 2$ distance among sites Vs Euclidean distances in a PCA.

CORRESPONDENCE ANALYSIS (CA)

Practical Points:

- All the used **categorical** data should be in the same scale.
  - CA must be computed on raw data → **DO NOT TRANSFORM!**
- CA produces one less axis than variables.
- Orthogonal axes are ranked in decreasing order.
- The focus **could** be
  - The sites (usually rows) → this is called `scaling =1`
  - The variables (usually columns) → this is called `scaling =2`
- Just like a PCA, The broken stick model can be used as guidance to the number of axes to retain.

CORRESPONDENCE ANALYSIS (CA)
IMPLEMENTING IT IN R

An automatic version of CA in R is implemented in as part of many packages:

- `ca()` → from the `ca` package
- `CA()` → from the `FactoMineR` package
- `cca()` → from the `vegan` package

This is my preferred option

CORRESPONDENCE ANALYSIS (CA)

IMPLEMENTING IT IN R

```
> require(vegan)
> data(varespec)
> CA.ord = cca(varespec)
> CA.ord
Call: cca(X = varespec)
              Inertia Rank
Total           2.083
Unconstrained   2.083   23
Inertia is scaled Chi-square

Eigenvalues for unconstrained axes:
   CA1    CA2    CA3    CA4    CA5    CA6    CA7    CA8
0.5249 0.3568 0.2344 0.1955 0.1776 0.1216 0.1155 0.0889
(Showing 8 of 23 unconstrained eigenvalues)
```

The proportion explained is estimated by divining each Eigenvalue by the **Total inertia**

Unlike in a PCA these are **NOT** the amount of variation represented along an axis

CORRESPONDENCE ANALYSIS (CA)

HOW MANY AXES?

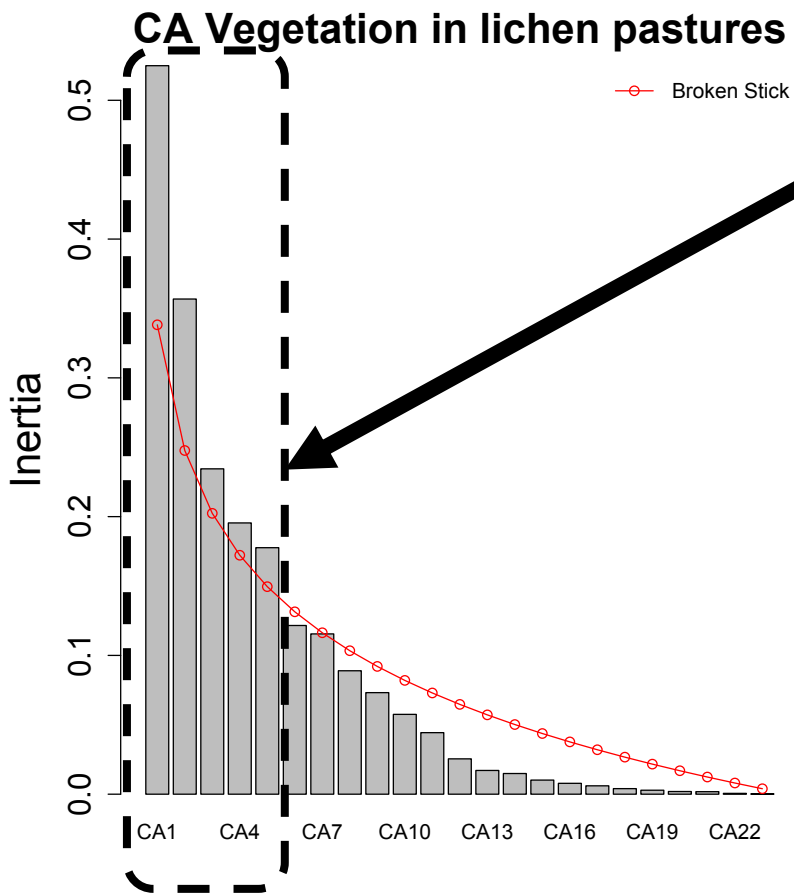
Amount of variation represented along an axis

```
data(varespec);CA.ord = cca(varespec)
# Extract the Amount of variation represented along an axis
summary(CA.ord)$ cont$ importance
Importance of components:
                          CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8   CA9
Eigenvalue              0.525 0.357 0.234 0.195 0.178 0.122 0.115 0.089 0.073

Proportion Explained  0.252 0.171 0.113 0.094 0.085 0.058 0.055 0.043 0.035

Cumulative Proportion 0.252 0.423 0.536 0.630 0.715 0.773 0.829 0.871 0.906
..
```

Like in a PCA the first axis captures most of the variance

You can also use the cumulative proportion to select the number of axes

CORRESPONDENCE ANALYSIS (CA)
HOW MANY AXES?



Broken stick approach

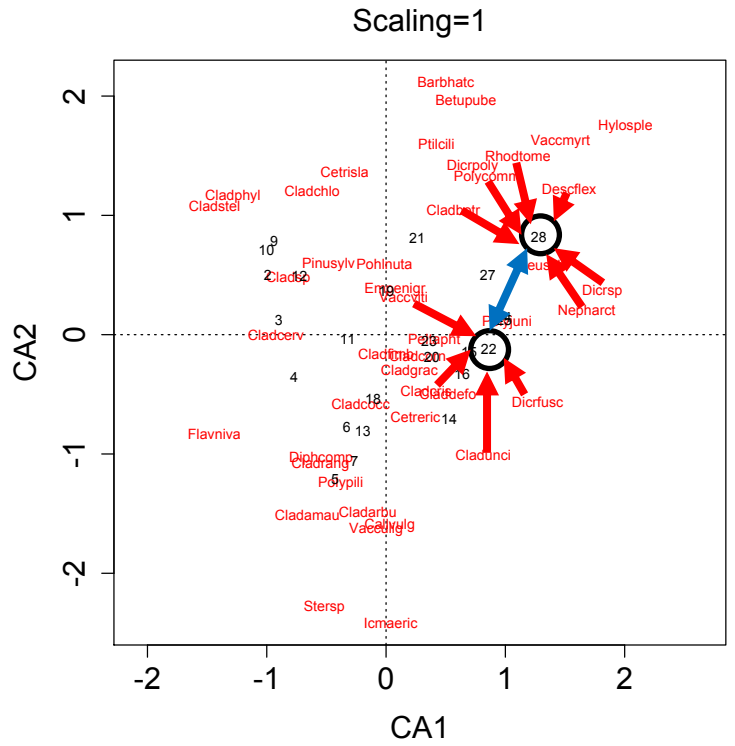
Observed var is larger than expected

The first 5 axes should be kept

```
data(varespec)
CA.ord = cca(varespec)
screeplot(CA.ord,
          bstick = TRUE,
          npcs = length(CA.ord $CA$eig),
          main="CA Vegetation in lichen pastures",
          cex.lab=2,cex.axis=1.5,cex.main=2)
```

CORRESPONDENCE ANALYSIS (CA)
VISUALISING AN CA

- In scaling 1 biplots, the sites are at chi-square distances of one another (blue arrows)
- The contributions of the species to the sites are reflected by the site-to-species distances (red arrows).
- The species are around the sites, in positions reflecting their abundances at each site.



Scaling 1
Appropriate to look the ordination of Sites

↓

Sites are centroids of species

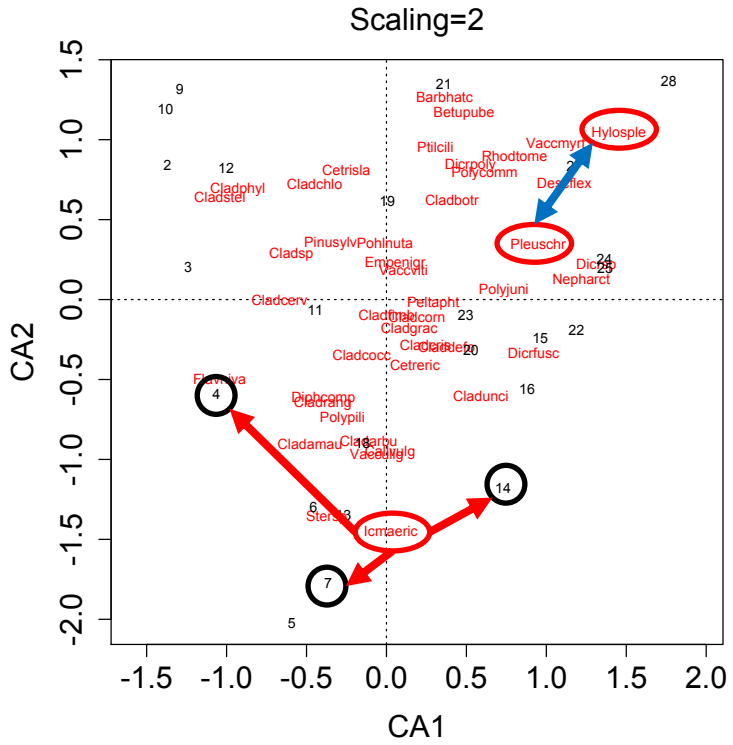
CORRESPONDENCE ANALYSIS (CA)

VISUALISING AN CA

Scaling 2
Appropriate to look the ordination of Species

↓

Species are centroids of Sites



- In scaling 2 biplots, the species are at chi-square distances of one another (Blue arrow).
- The sites are around the species, in positions reflecting species abundances at the sites (red arrows).
  - Species proximity to a site means it is more abundant there.

So far so good?

Any questions?

Ready to move on?

DISSIMILARITY BASED APPROACH
PRINCIPLE COORDINATES ANALYSES

The objective of Principal coordinate analysis (PCoA) is

- to represent the evaluated objects in a full-dimensional Euclidean space.
- by fully **reconstruct the original dissimilarities** among the objects in a reduced and orthogonal space.
  - Think of it as a transformation after the computation of an appropriately chosen dissimilarity measure.
- Like PCA and CA, PCoA produces a set of orthogonal axes whose importance is measured by eigenvalues.
- This is sometimes called classical multidimensional scaling.

PCA, CA, AND PCOA
WHAT IS THE DIFFERENCE?

A **PCoA** differs from a PCA in that a PCA goal is to preserve the **correlation/covariation** of objects.

A **PCoA** differs from a CA in that a CA goal is preserving the **Chi-Squared distance** among the objects while.

PCoA goal is preserving the **ANY TYPE OF DISTANCE** among the objects.

PRINCIPLE COORDINATES ANALYSES
AN IMPORTANT POINT

The possibility of using *non-metric distances* (i.e., Bray Curtis, Gower's) means that there is a possibility of getting "**negative**" eigenvalues if the distance matrix is not **EUCLIDEAN**!

- **This is bad because** it could be so that these negative eigenvalues "overwrite" the variance captured by the first axes.

PRINCIPLE COORDINATES ANALYSES
AN IMPORTANT POINT

You can solve the "**negative**" eigenvalues problem by adding

- Use the squared root of the distance.

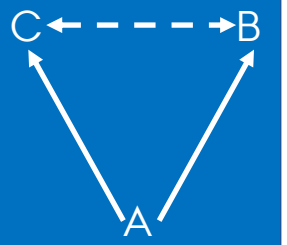
- Adding a constant to the squared distances ➔ *Lingoes correction.*

- Adding a constant to the raw distances ➔ *Cailliez correction*

PRINCIPLE COORDINATES ANALYSES
IS MY DISTANCE EUCLIDEAN

```
> require(vegan)
> require(ade4)
> data(mite)
> mat1  = as.matrix(mite[1:10, 1:15])

> #Example 1: compute Bray Curtis distance for mat1
> spe.bray <- vegdist(mat1)
> ade4::is.euclid(spe.bray)
[1] FALSE

> #Example 2: Cailliez correction
> ade4::is.euclid(spe.bray+1)
[1] TRUE
> #Example 3: Lingoes correction-
> ade4::is.euclid((spe.bray^2)+1)
[1] TRUE
```

A reminder:
Euclidean property means that the distance follows the Pythagoras formula.



In this case the distances for this community data are **NOT** Euclidean

In these cases the corrections make the community data Euclidean

PRINCIPLE COORDINATES ANALYSES
IS MY DISTANCE EUCLIDEAN

```
require(vegan)
require(adespatial)
data(mite)
mat1  = as.matrix(mite[1:10, 1:15])   # No column has a sum of 0

> #Example 1: compute Hellinger distance for mat1
> D.out = adespatial::dist.ldc(mat1, method = "hellinger")
Info -- This coefficient is Euclidean
> ade4::is.euclid(D.out)
[1] TRUE

> #Example 2: compute percentage difference dissimilarity for
mat2
> D.out = adespatial::dist.ldc(mat1, method = "percentdiff")
Info -- For this coefficient, sqrt(D) would be Euclidean
> ade4::is.euclid(D.out)
[1] FALSE
> ade4::is.euclid(sqrt(D.out))
[1] TRUE
```

In this case the distances for this community data are Euclidean

In this case the distances for this community data are **NOT** Euclidean but becomes Euclidean after the suggested transformation

PRINCIPLE COORDINATES ANALYSES
IMPLEMENTING IT IN R

Like PCA/CA there are multiple options

- `cmdscale` from the package `vegan`
- `pco` from the package `ecodist`
- `pcoa` from the package `ape`

All these use as an input a distance matrix

My preferred option

PRINCIPLE COORDINATES ANALYSES
IMPLEMENTING IT IN R

Like PCA there are multiple options

- `cmdscale` from the package `vegan`
- `pco` from the package `ecodist`
- `pcoa` from the package `ape`

Can directly specify a correction methods for negative eigenvalues

PRINCIPLE COORDINATES ANALYSES
IMPLEMENTING IT IN R

```
spe.bray.LinCor <- 1+vegdist(varespec)^2
ade4::is.euclid(spe.bray.LinCor)
pcoa.V1 <- cmdscale(spe.bray.LinCor,
                    k = (nrow(varespec) - 1),
                    eig = TRUE)

str(pcoa.V1)
List of 5
 $ points: num [1:24, 1:23] 0.177 -0.517 -0.624 -0.585 -0.314
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:24] "18" "15" "24" "27" ...
  .. ..$ : NULL
 $ eig   : num [1:24] 5.42 3.62 1.59 1.42 1.1 ...
 $ x     : NULL
 $ ac    : num 0
 $ GOF   : num [1:2] 1 1
```

The positions in a PCoA space with 24 dimensions (the same number of sites.

The Eigenvalues.

PRINCIPLE COORDINATES ANALYSES
IMPLEMENTING IT IN R

```
spe.bray.LinCor <- 1+vegdist(varespec)^2
ade4::is.euclid((spe.bray.LinCor)+1)
pcoa.V2 <- ecodist::pco(spe.bray.LinCor)

str(pcoa.V2)

List of 2
 $ values : num [1:24] 5.42 3.62 1.59 1.42 1.1 ...
 $ vectors: num [1:24, 1:24] 0.0326 -0.0954 -0.1152 -0.108 -0.058 ...
 - attr(*, "class")= chr "eigen"
```

The Eigenvalues.

The positions in a PCoA space with 24 dimensions (the same number of sites.

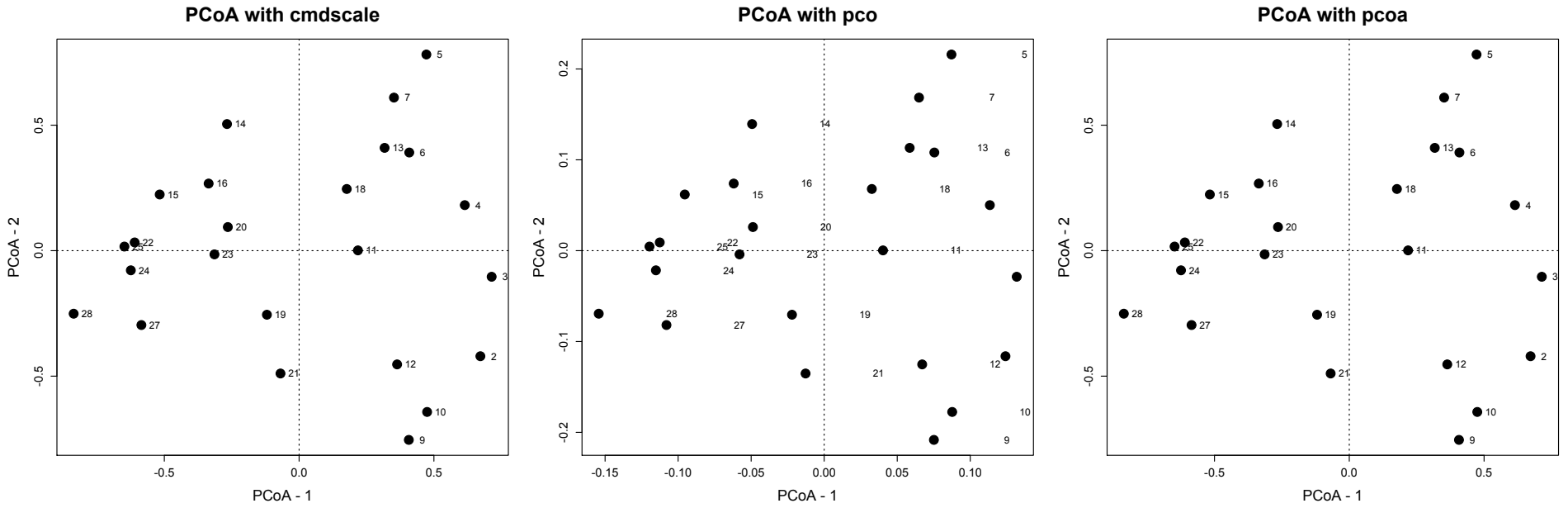
PRINCIPLE COORDINATES ANALYSES
IMPLEMENTING IT IN R

```
spe.bray.LinCor <- 1+vegdist(varespec)^2
ade4::is.euclid((spe.bray.LinCor)+1)
pcoa.V3 <- ape::pcoa(spe.bray.LinCor)
str(pcoa.V3)
List of 5
 $ correction: chr [1:2] "none" "1"
 $ note      : chr "There were no negative eigenvalues. No correction was
applied"
 $ values    : 'data.frame': 23 obs. of 5 variables:
  ..$ Eigenvalues   : num [1:23] 5.42 3.62 1.59 1.42 1.1 ...
  ..$ Relative_eig  : num [1:23] 0.2371 0.1584 0.0696 0.062 0.048 ...
  ..$ Broken_stick  : num [1:23] 0.1624 0.1189 0.0971 0.0827 0.0718 ...
  ..$ Cumul_eig     : num [1:23] 0.237 0.395 0.465 0.527 0.575 ...
  ..$ Cumul_br_stick: num [1:23] 0.162 0.281 0.378 0.461 0.533 ...
 $ vectors   : num [1:24, 1:23] 0.177 -0.517 -0.624 -0.585 -0.314 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:24] "18" "15" "24" "27" ...
  .. ..$ : chr [1:23] "Axis.1" "Axis.2" "Axis.3" "Axis.4" ...
 $ trace     : num 22.9
 - attr(*, "class")= chr "pcoa"
```

The Eigenvalues.

The positions in a PCoA space with 24 dimensions (the same number of sites.

PRINCIPLE COORDINATES ANALYSES
IMPLEMENTING IT IN R



The proximity between points represents how similar is the composition of these

All three approaches return the same ordination and eigen values but the scaling is different

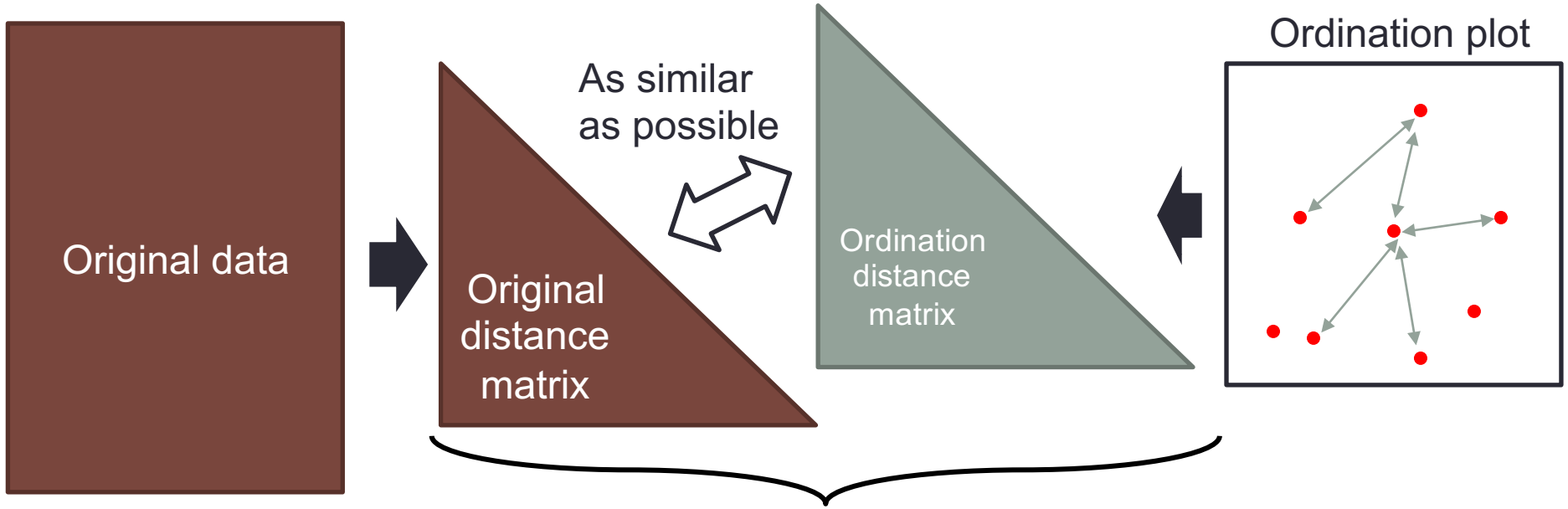
So far so good?

Any questions?

Ready to move on?

(NON) METRIC MULTIDIMENSIONAL SCALING – (N)MDS

Like PCoA, the goal here is visualization that maintains the original distances as well as possible



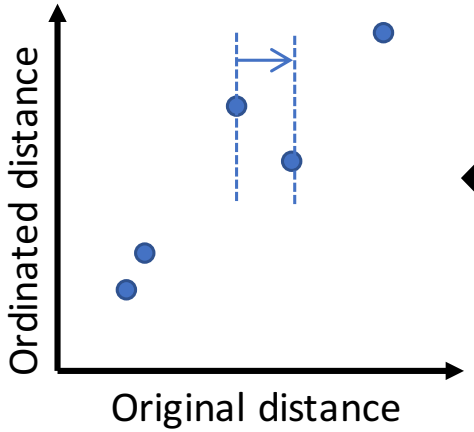
This transformation aims to rescale the original differences **into a predefine number of axes** by minimizing the stress

METRIC (NONMETRIC) MULTIDIMENSIONAL SCALING

What is stress?→ Squared differences between **original** and **new** distances

$$\sum \left(\begin{array}{cccccccc} 0 & 1 & 3 & 2 & 7 & 2 & 1 & 5 \\ & 0 & 5 & 9 & 3 & 4 & 1 & 2 \\ & & 0 & 8 & 1 & 2 & 1 & 6 \\ & & & 0 & 6 & 7 & 8 & 0 \\ & & & & 0 & 2 & 1 & 4 \\ & & & & & 0 & 5 & 6 \\ & & & & & & 0 & 5 \\ & & & & & & & 0 \end{array} - \begin{array}{cccccccc} 0 & 2 & 5 & 4 & 7 & 2 & 5 & 7 \\ & 0 & 5 & 2 & 3 & 1 & 6 & 4 \\ & & 0 & 3 & 2 & 4 & 1 & 5 \\ & & & 0 & 6 & 7 & 8 & 2 \\ & & & & 0 & 2 & 1 & 3 \\ & & & & & 0 & 4 & 7 \\ & & & & & & 0 & 1 \\ & & & & & & & 0 \end{array} \right)^2$$

How to visualize stress?:



Shepard diagram *visualizes the "Stress"*

(N)MDS - THE BASICS

Metric multidimensional scaling

- Optimization procedure to a variety of loss functions (i.e., *stress*) which is often minimized using a procedure called **stress majorization**.

Non-metric multidimensional scaling

- Optimization procedure based on a non-parametric monotonic relationship between the dissimilarities in the item-item matrix and the Euclidean distances between items

AN IMPORTANT POINT (N)MDS

The first axis is not necessarily the most important

- The first axis of a 2D (N)MDS is NOT the same as the first axis of a 3D (N)MDS
  - The number of axes you choose at the beginning matters!
    - This is an argument in the function → more axes reduces the stress.

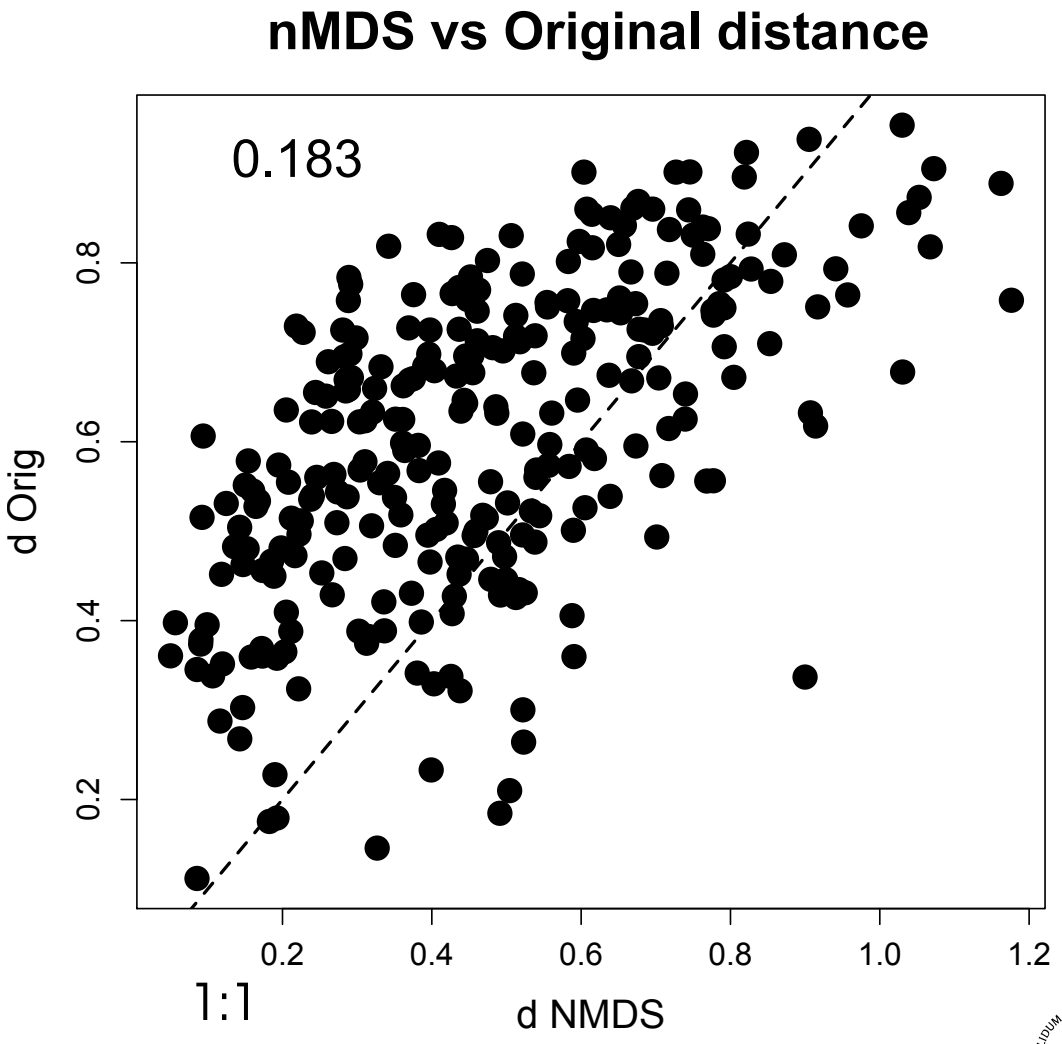
  - (N)MDS'ing your data twice could produce different results as the optimisation done using iterative process not an analytical solution.

(N)MDS - IMPLEMENTING IT IN R

```
require(vegan)
data(varespec) # 24 sites and 44
species
mds = metaMDS(varespec,
              distance = "bray",
              k=2)
```

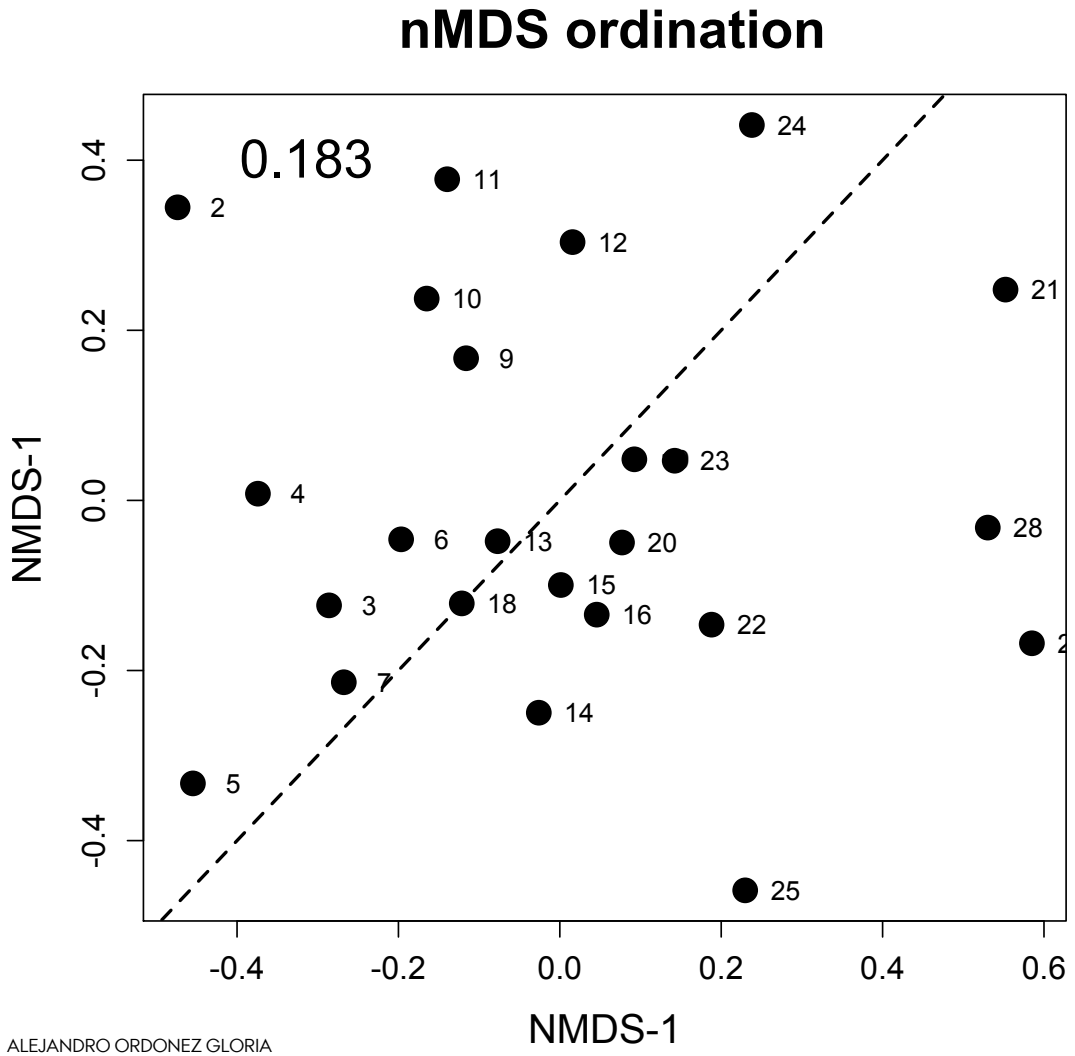
Alternative functions:

`metaMDS()` in `vegan` package
`NMDS()` in `ecodist` package
`isoMDS()` in `MASS` package



(N)MDS - EXTRACTING THE USEFUL INFORMATION

```
require(vegan)
data(varespec) # 24 sites and 44 species
mds = metaMDS(varespec,
              distance = "bray",
               k=2)
> mds$stress
[1] 0.1825658
> mds$points
             MDS1          MDS2
18 -0.121503616 -0.121104640
15  0.001263504 -0.099472058
24  0.238016961  0.441366552
...
```



So far so good?

Any questions?

Ready to finish?

IN SUMMARY...

Ordination is about ***arranging*** multivariate objects

- **Based on correlation/covariances:** PCA
- **Based on distances:** CA, PCoA, NMDS.

These Methods

- Identify the primary axes of variation in data and reduce the number of dimensions to a more manageable number.
- Focus on reorganizing the data to assess the existence of patterns in the evaluated objects.

AARHUS
UNIVERSITY