Procrustes, PCA, and 3D coordinates

Euclid

Outline

Procrustes superimposition

Measuring similarity and difference with morphometric distances

Procrustes distances (the standard in geometric morphometrics)

Ordination techniques

Principal Components Analysis (PCA)

Rigid rotation

Outputs of PCA: eigenvalues, eigenvectors and scores

Shape modeling

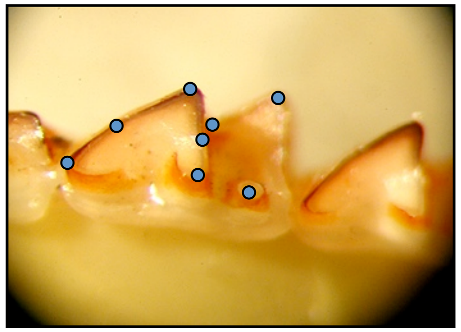
Procrustes superimposition

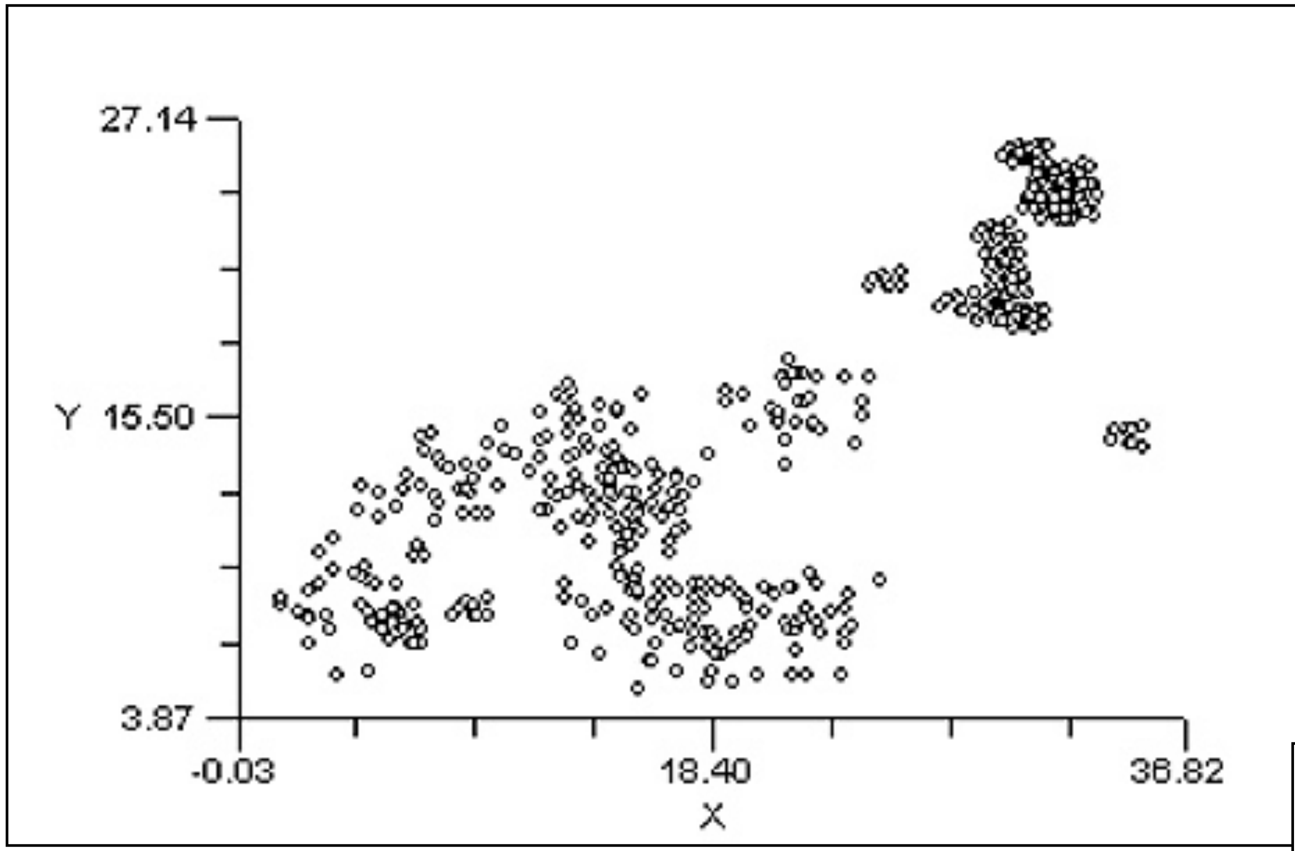
also known as...

Procrustes analysis
Procrustes fitting
Generalized Procrustes Analysis (GPA)
Generalized least squares (GLS)
Least squares fitting

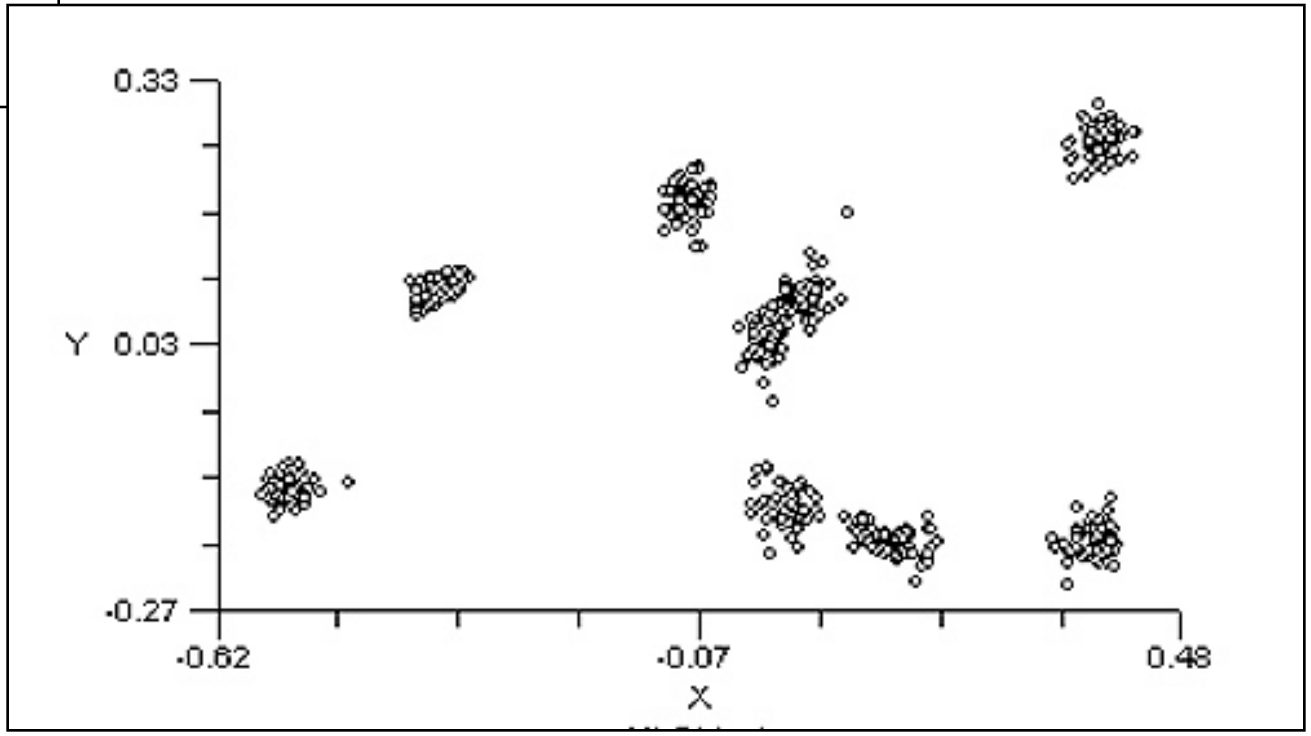
- Centers all shapes at the origin (0,0,0)
- Usually scales all shapes to the same size (usually “unit size” or size = 1.0)
- Rotates each shape around the origin until the sum of squared distances among them is minimized (similar to least-squares fit of a regression line)
- Ensures that the differences in shape are minimized

Procrustes superimposition





Before Procrustes

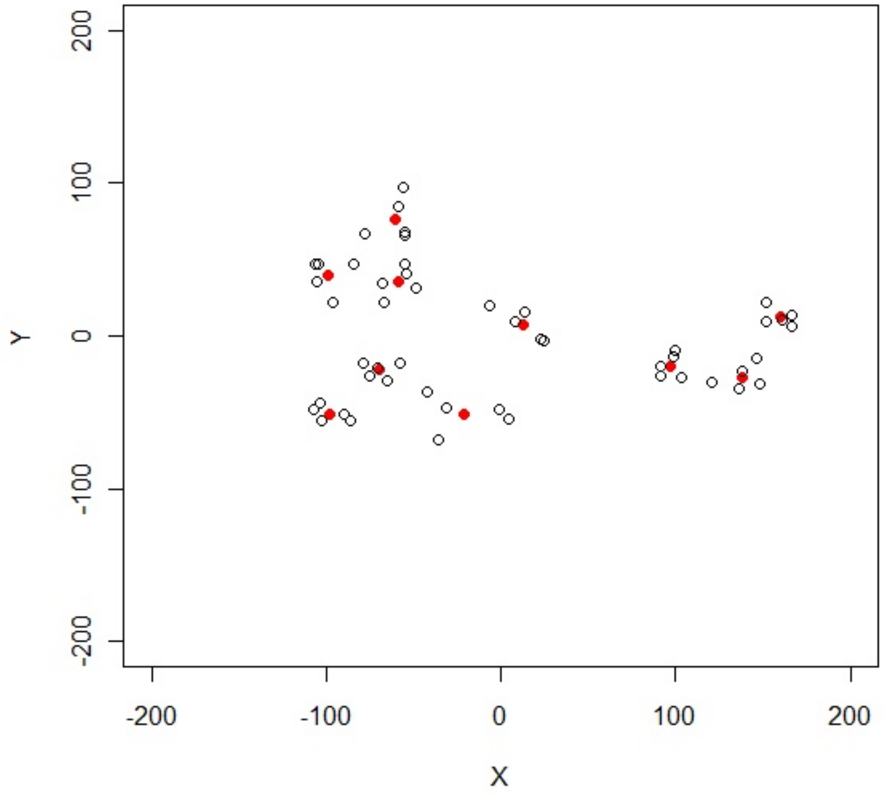


After Procrustes

Definitions

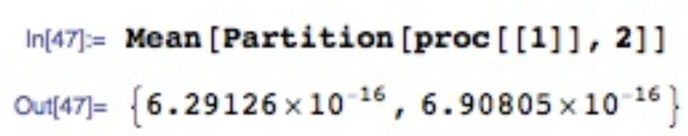
Shape - a configuration of landmarks irrespective of size ("form" is sometimes used to distinguish a set of landmarks that have a scale)

Consensus shape - the mean shape of a sample

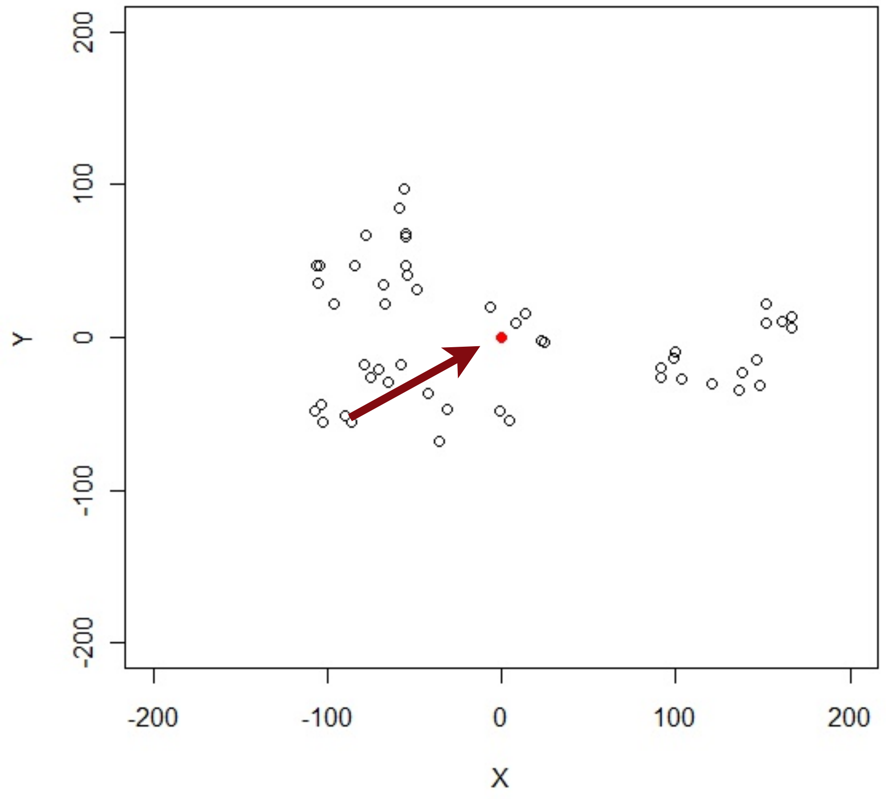


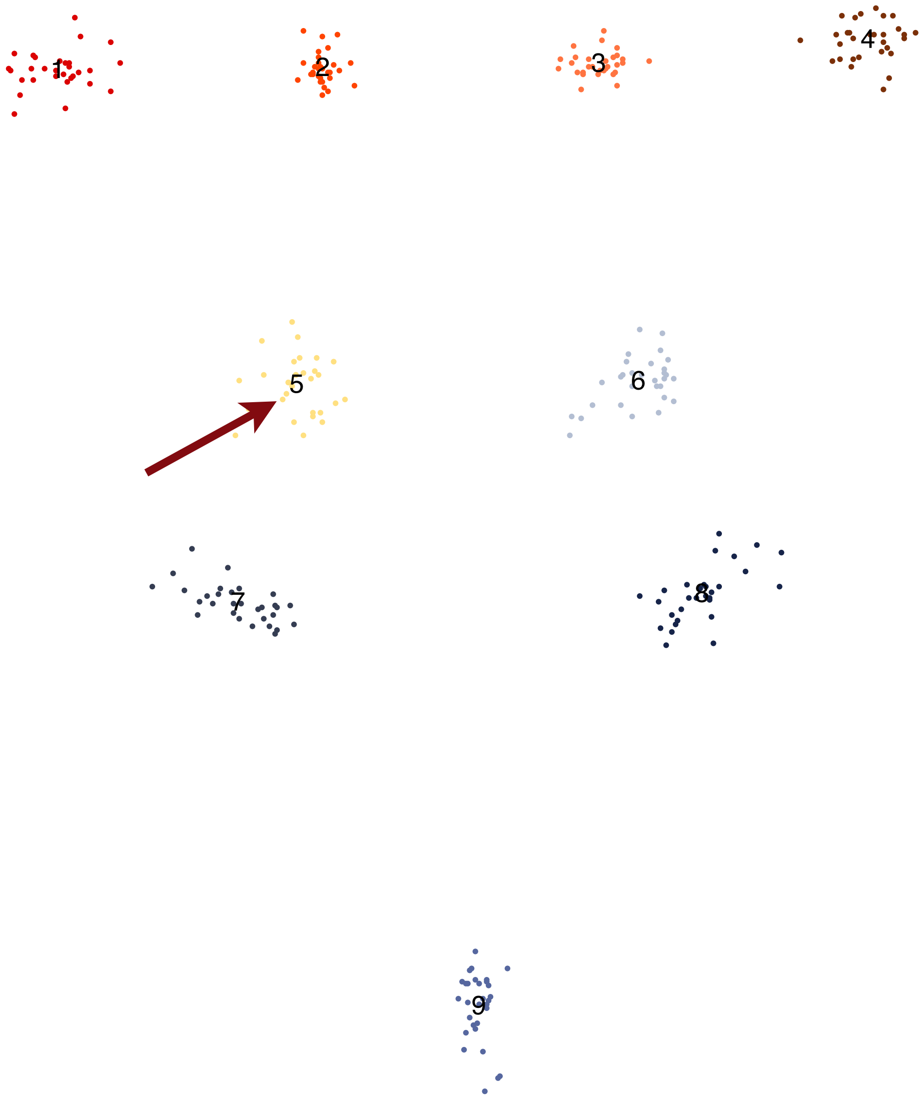
More definitions

Centroid - the center point of a shape, a sample of shapes, or a single landmark in a sample of shapes



```
In[47]:= Mean[Partition[proc[[1]], 2]]
Out[47]= {6.29126×10^-16, 6.90805×10^-16}
```

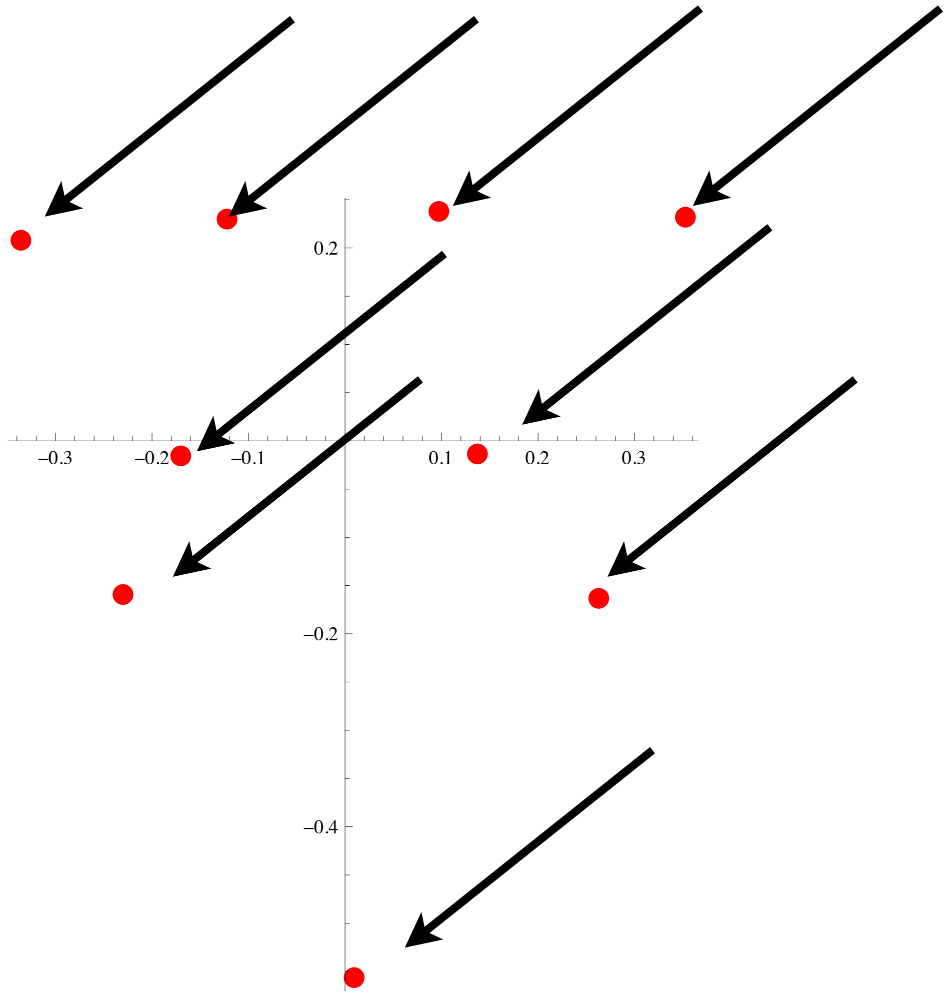




Procrustes: Translation, Rotation, Rescaling

Thought problem...

How do you translate one of your original landmark shapes to the point {0,0} on a graph?



Goal of morphometrics: measuring similarity and difference

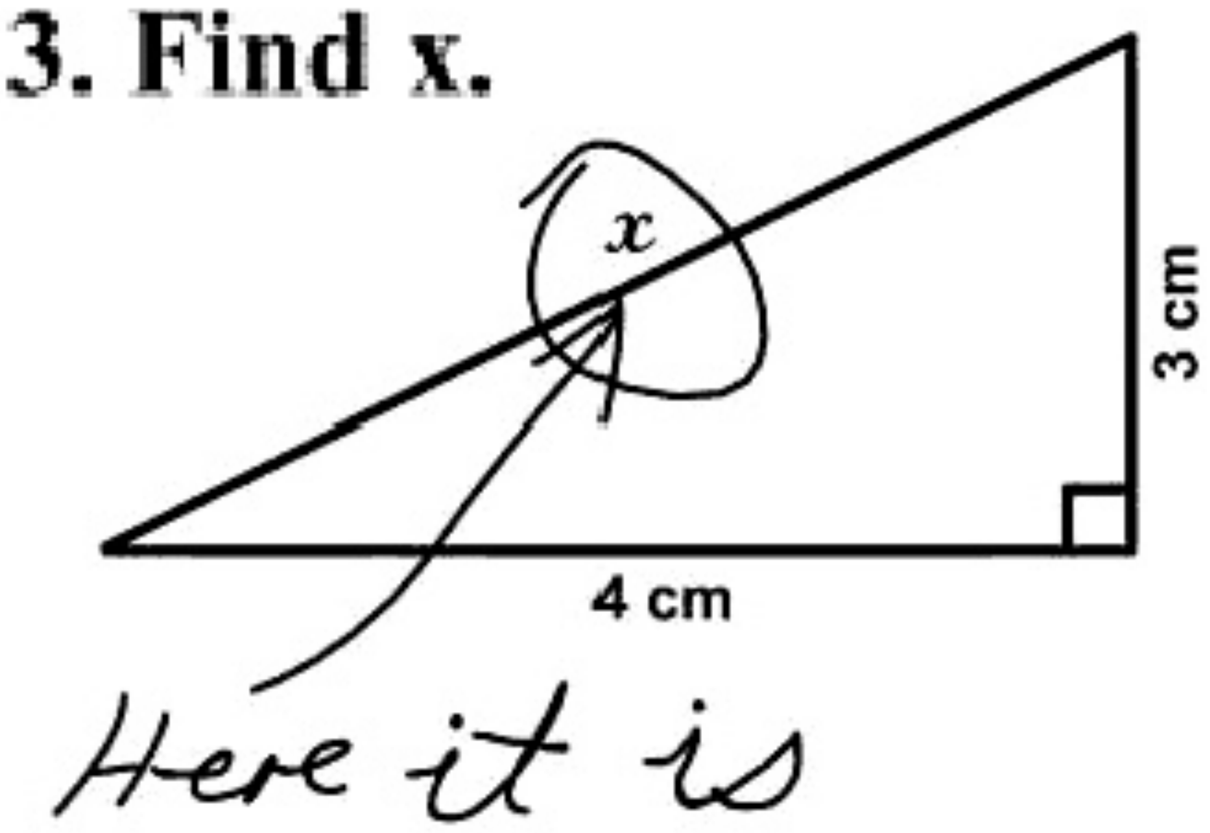
Morphometric distances are the main measure of difference

Measured as the difference between objects (which might be specimens or means of species, or whatever) on all the variables being used

In geometric morphometrics, the main measure of difference is the Procrustes distance, the distance between shapes after they have been superimposed

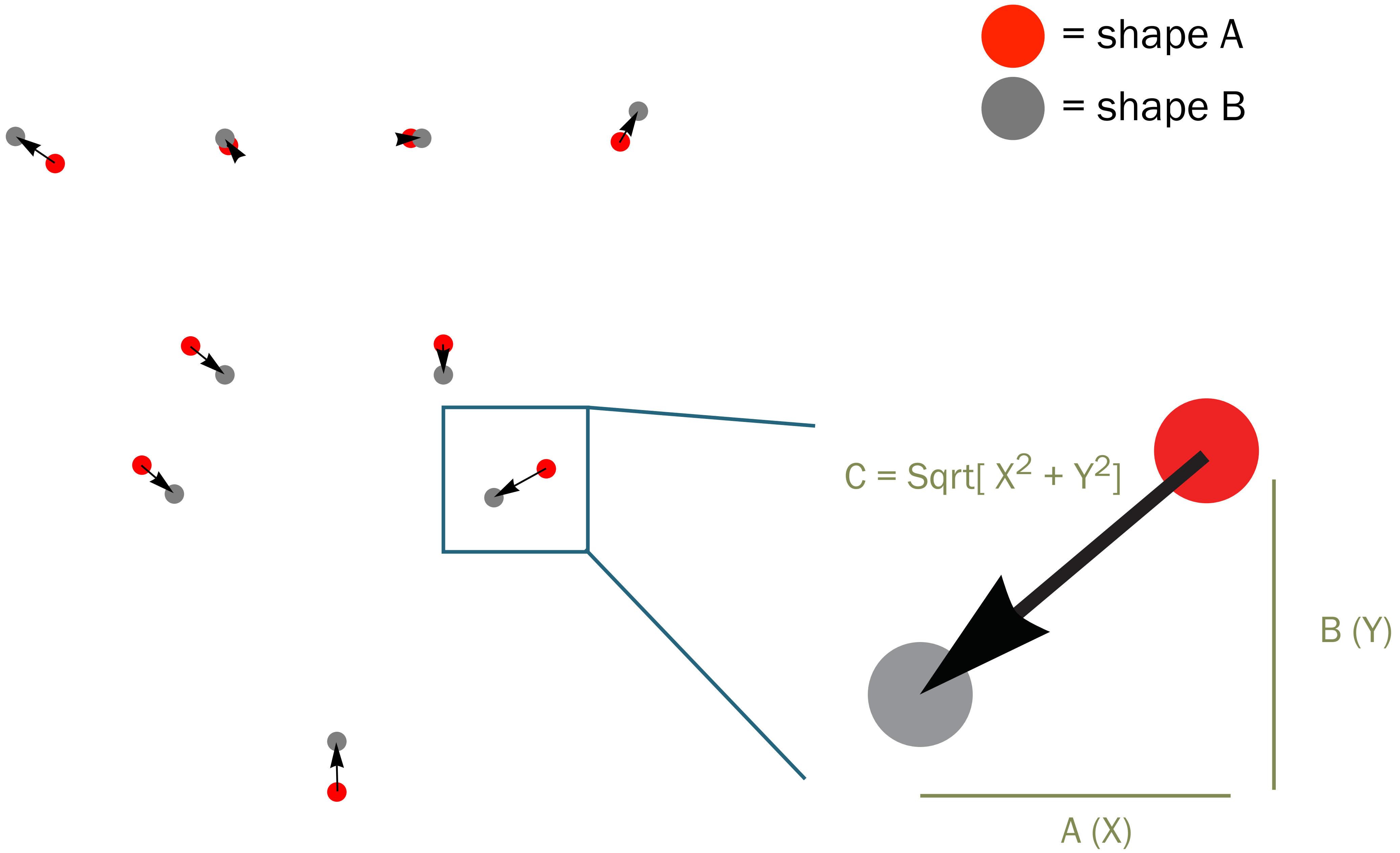
Morphometric distances = Euclidean distances (more or less)

Euclidean distance = sqrt (a^2 + b^2)



Procrustes distance

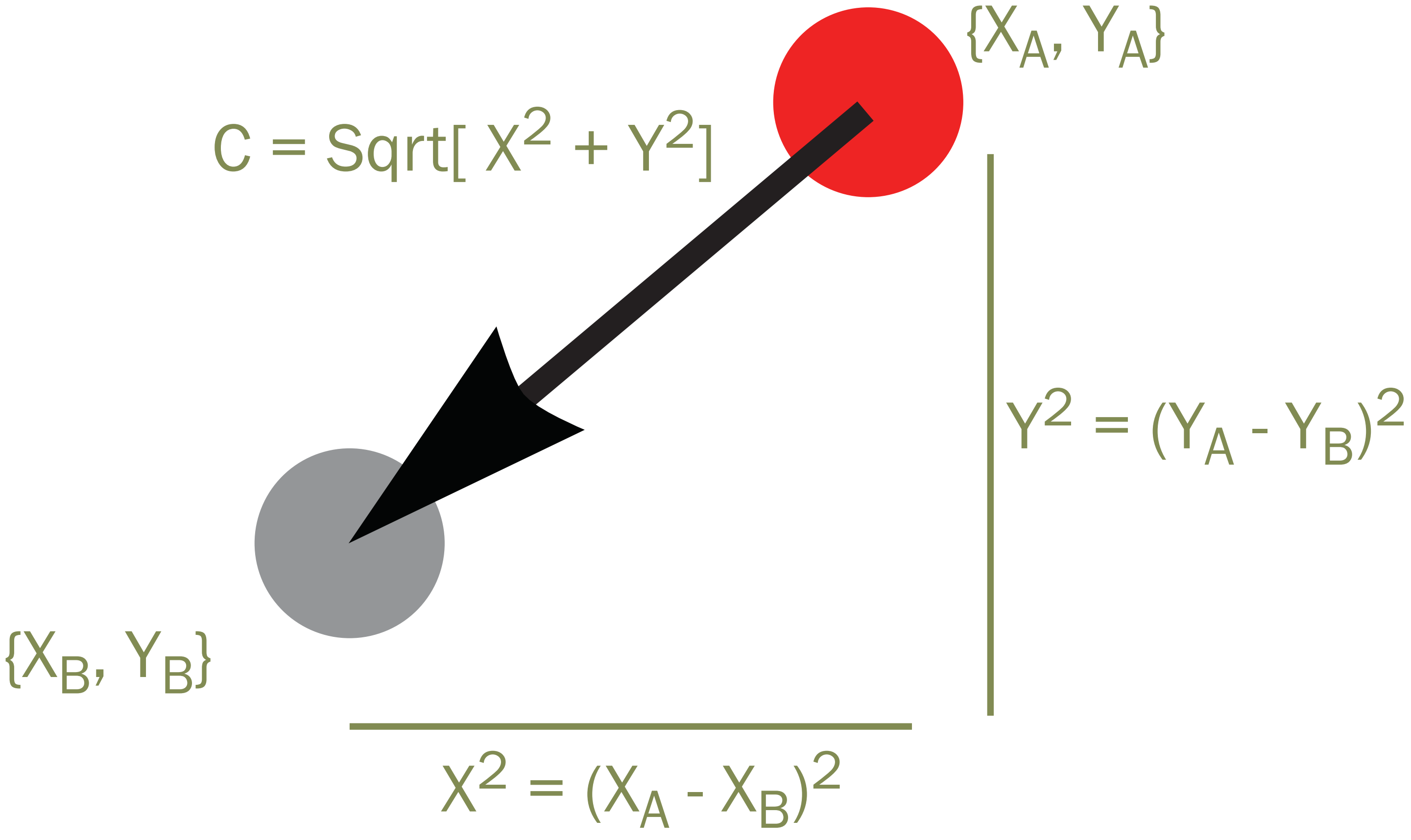
Sum of the distances between corresponding landmarks of two shapes.



Procrustes distance in Mathematica

How would you calculate the Procrustes distance between the mean shape (consensus) and any one of the objects?

1. How to calculate X^2? Y^2?



$\{X_A, Y_A\}$

$C = \text{Sqrt}[\ X^2 + Y^2]$

$Y^2 = (Y_A - Y_B)^2$

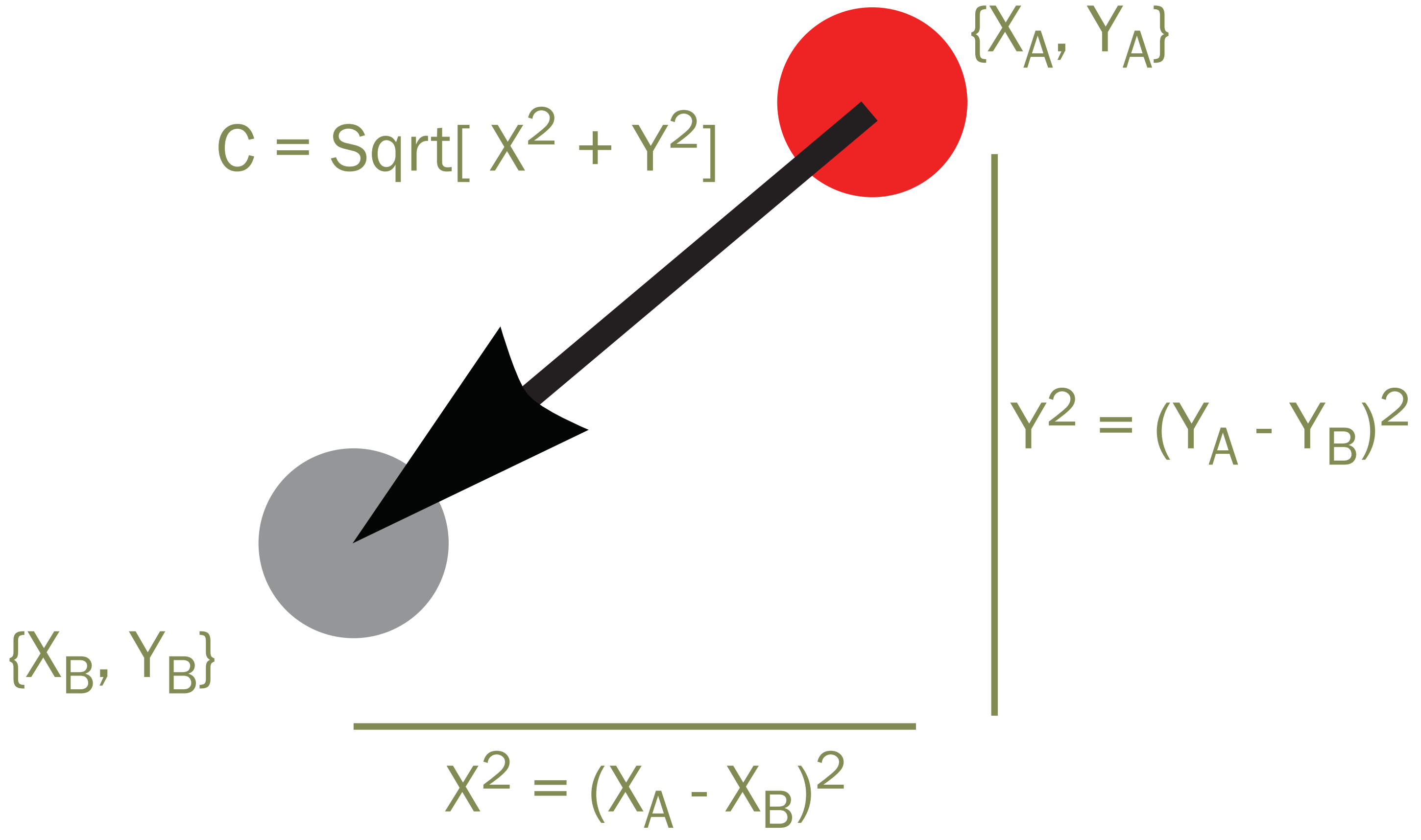
$\{X_B, Y_B\}$

$X^2 = (X_A - X_B)^2$

Procrustes distance in Mathematica

How would you calculate the Procrustes distance between the mean shape (consensus) and any one of the objects?

1. How to calculate X^2? Y^2?



$\{X_A, Y_A\}$

$C = \text{Sqrt}[\ X^2 + Y^2]$

$Y^2 = (Y_A - Y_B)^2$

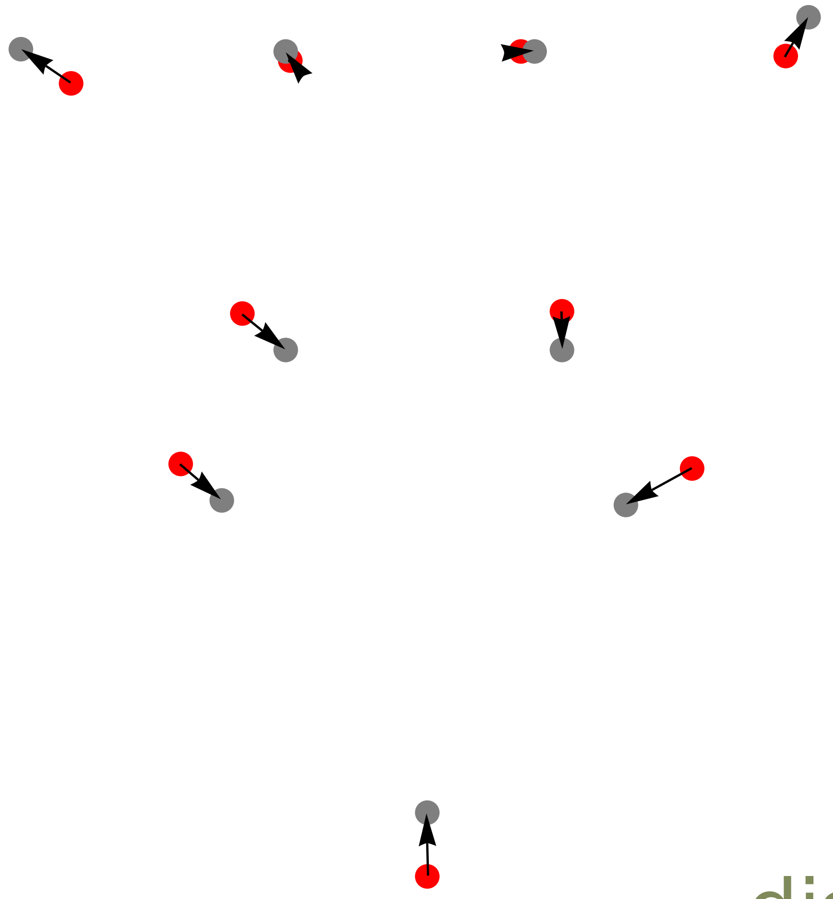
$\{X_B, Y_B\}$

$X^2 = (X_A - X_B)^2$

$(\{X_A, Y_A\} - \{X_B, Y_B\})$ ^2

Procrustes distance

Sum of the distances between corresponding landmarks of two shapes.



🔴 = shape A

⚫ = shape B

In Mathematica:

1. subtract entire line of coordinates of B from A
2. square them
3. sum them
4. take the square root

```
dist = Sqrt[ Plus@@((A-B)^2)]
```

Note: *Plus@@* sums whatever comes after it

Use Procrustes distance to find outliers

Q. Which faces are most different from the average?

A. The ones most distant from the mean shape (consensus).

In Mathematica:

```
1.    consensus = Mean[proc]

2.    dists = Table[Sqrt[Plus @@ ((proc[[x]] - consensus)^2)], {x, Length[proc]}]

3.    Sort[Transpose[{dists, labels}]]
```

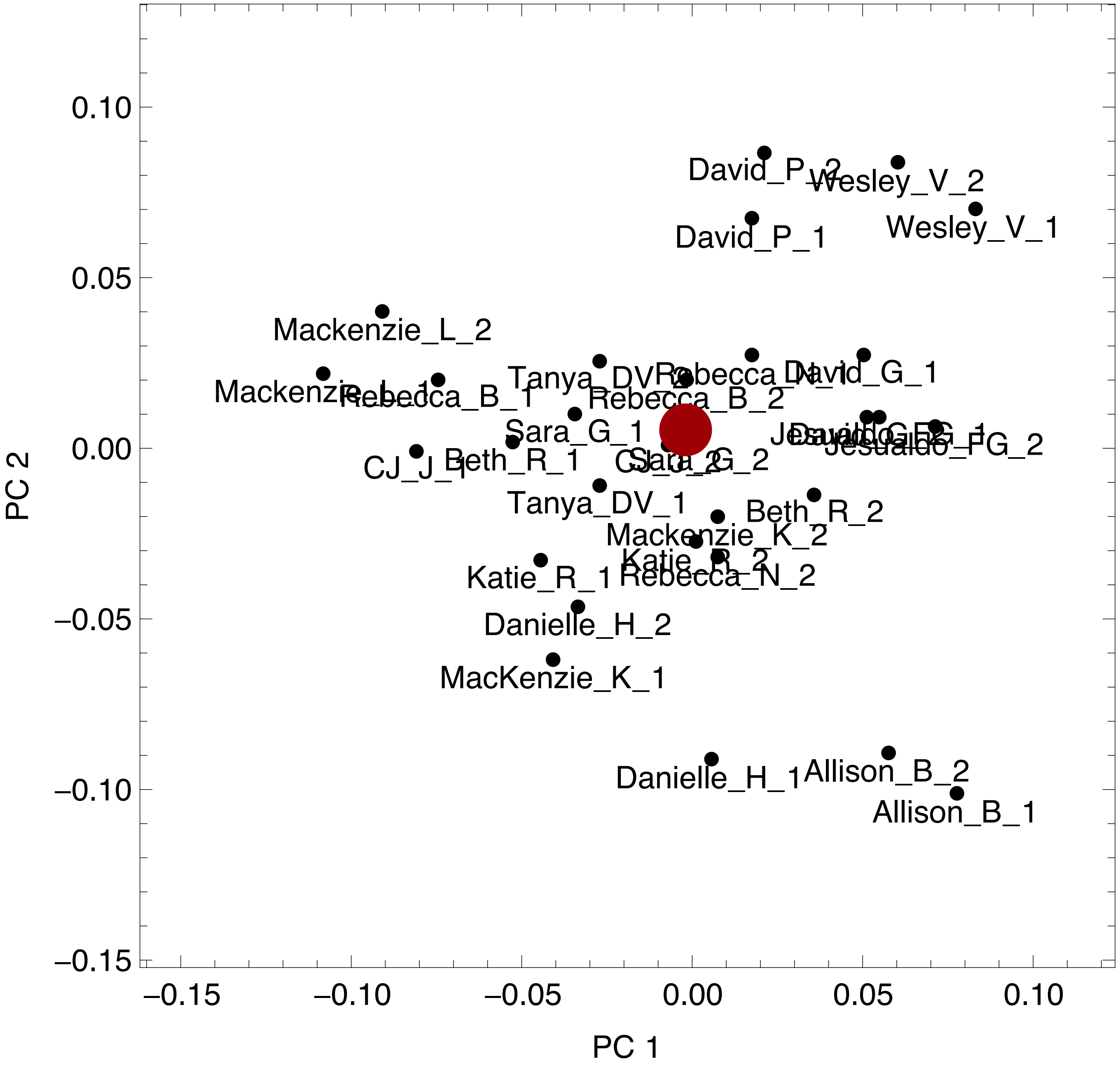
```
{{0.0457236, "Rebecca_B_2"}, {0.0493794, "Tanya_DV_2"}, {0.0515922, "Rebecca_N_1"}, {0.0541266,
"Beth_R_2"}, {0.0570841, "Katie_R_2"}, {0.058025, "Sara_G_2"}, {0.0581106, "CJ_J_2"}, {0.0608825,
"David_G_2"}, {0.060957, "Beth_R_1"}, {0.0611666, "Sara_G_1"}, {0.0655094, "Rebecca_N_2"}, {0.0720819,
"Mackenzie_K_2"}, {0.0778101, "Tanya_DV_1"}, {0.0781057, "Danielle_H_2"}, {0.0784817, "Jesualdo_FG_1"},
{0.078695, "David_G_1"}, {0.0818828, "Jesualdo_FG_2"}, {0.0854001, "MacKenzie_K_1"}, {0.0871498,
"David_P_1"}, {0.089637, "CJ_J_1"}, {0.0933814, "Katie_R_1"}, {0.104911, "Mackenzie_L_2"}, {0.10901,
"Wesley_V_2"}, {0.109114, "David_P_2"}, {0.111641, "Allison_B_2"}, {0.113724, "Rebecca_B_1"}, {0.120251,
"Wesley_V_1"}, {0.123681, "Mackenzie_L_1"}, {0.134277, "Allison_B_1"}, {0.134961, "Danielle_H_1"}}
```

Alternate function: ProcrustesDistance[A, B, *ndims*]

Why??

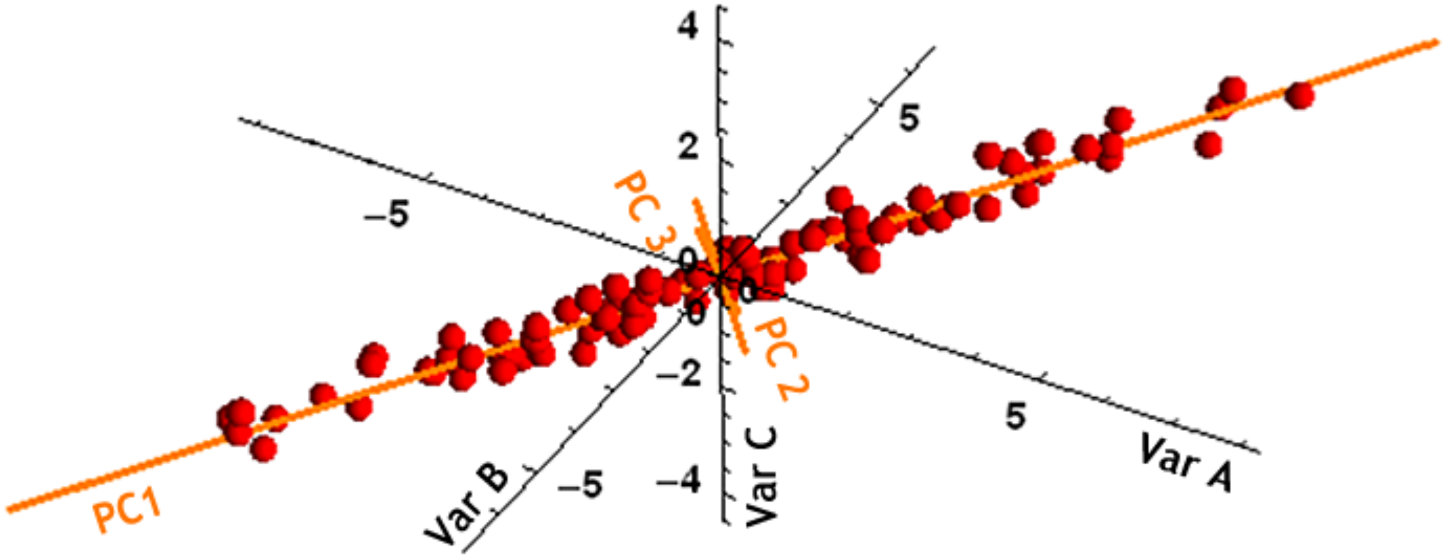
Compare reverse sorted list to PCA plot

0.135166	Danielle_H_1
0.134481	Allison_B_1
0.123859	Mackenzie_L_1
0.120421	Wesley_V_1
0.113878	Rebecca_B_1
0.11179	Allison_B_2
0.109257	David_P_2
0.109152	Wesley_V_2
0.105042	Mackenzie_L_2



Ordination and Principal Components Analysis

1. Introduction to Ordination
2. Why PCA is an important part of Geometric Morphometrics
3. Technical explanation of what PCA does
4. Eigenvalues, Eigenvectors and Scores
5. Morphological meaning of principal component axes
6. Modeling in shape space



Ordination

Ordering specimens along new variables

Principal Components Analysis (PCA)
Arranges data by major axes based on measured variables

Principal Coordinates Analysis (PCO)
Arranges data by major axes based on distance measures

Canonical Variates Analysis (CVA)
(or Discriminant Function Analysis, DFA)
Finds best separation between groups

Non Metric Multidimensional Scaling (NMDS)
Arranges data so the distances on 2D plot are as similar as possible to original multivariate distances

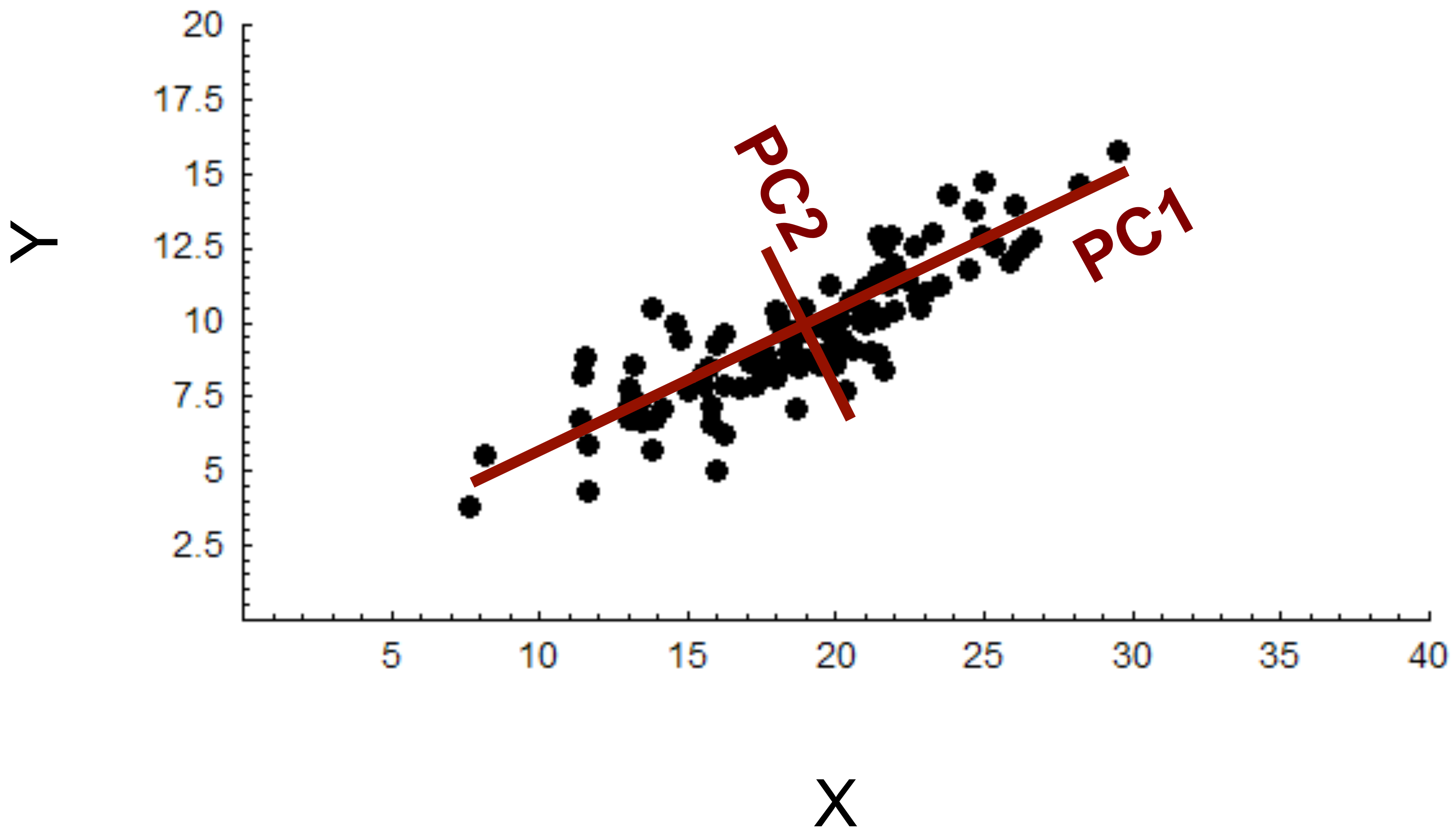
What does PCA do?

1. Rotates data to its major axes for better visualization

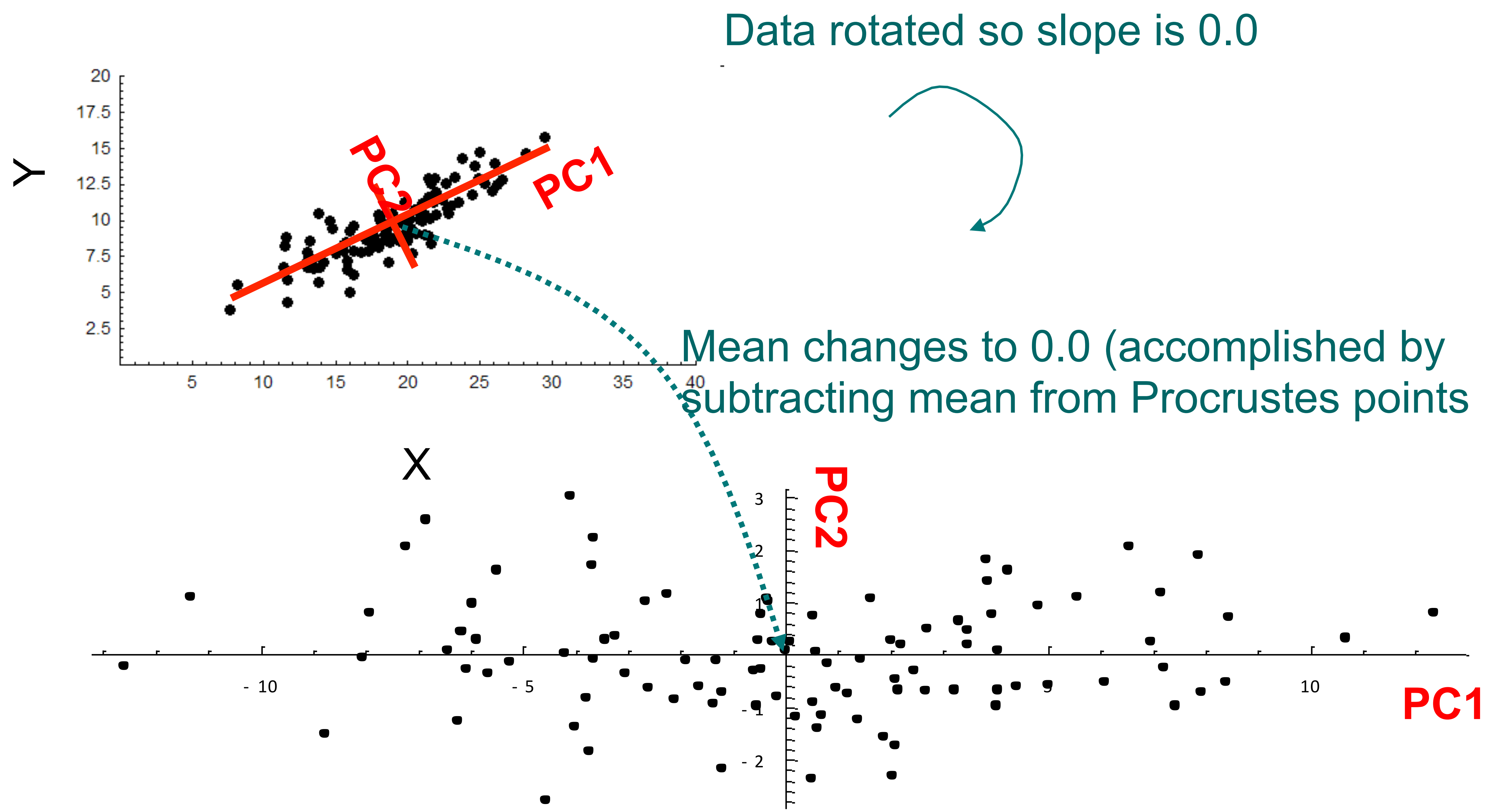
2. Preserves original distances between data points
(in other words, PCA does not distort the variation data, but only if the covariance method is used, which is standard in geometric morphometrics)

3. Removes correlations between variables to make further statistical analysis simpler

The principal components (PCs) of a data set are its major axes



Principal components are a ‘rigid rotation’ of the original data



Data rotated so slope is 0.0

Mean changes to 0.0 (accomplished by subtracting mean from Procrustes points

Note that variance increases along horizontal axis, but decreases along vertical axis.

Important points: the “meaning” of PCA

1. Principal components analysis finds the axes of greatest variation in a data set

2. PCA removes correlations from the data

3. Principal components scores are “shape variables” that are the basis for further analysis

4. But PCA is nothing more than a rotation of the data!

Behind the scenes in PCA of landmarks

Procrustes

This aligns shapes and minimizes differences between them to ensure that only real shape differences are measured.

1. **Subtract mean (consensus) from each shape to produce “residuals”**

   This centers the PC axes on the mean (consensus) shape.

2. **Calculate covariance matrix of residuals**

   Estimates variance and covariance among the original variables

3. **Calculate eigenvalues and eigenvectors of covariance matrix**

   Finds the major axes of the data and the variation along them.

4. **Multiply residuals times eigenvectors to produce scores**

   Rotates the original data onto the major axes and gives the coordinates for their new position.

Output of PCA

Eigenvalues

variance on each PC axis
(In Mathematica: *Eigenvalues[CM]*)

Eigenvectors

loading of each original variable on each PC axis
(In Mathematica: *Eigenvectors[CM]*)

Scores (=shape variables)

location of each data point on each PC axis
(In Mathematica: *PrincipalComponents[resids]*)

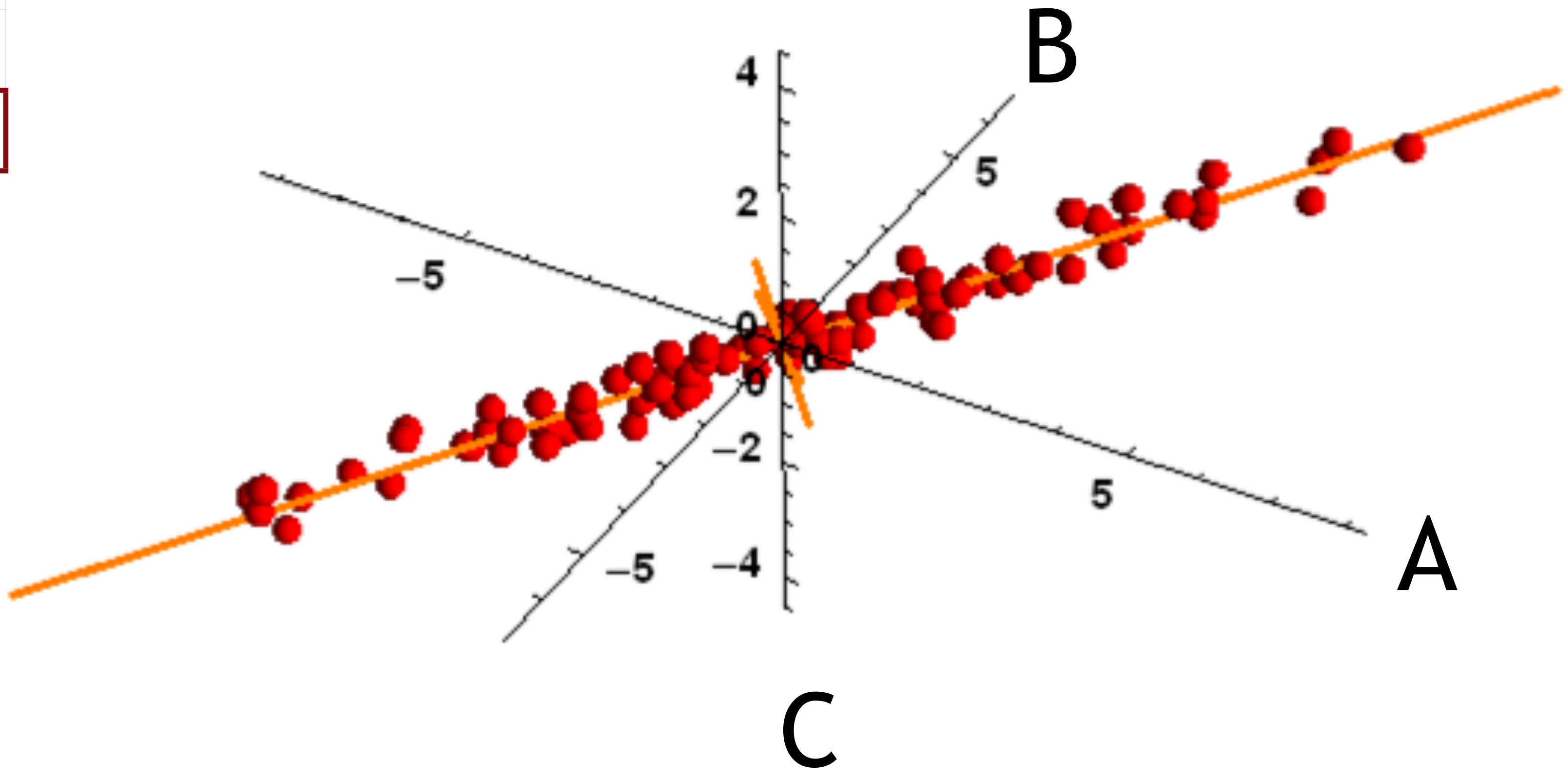
resids are the residuals of the Procrustes coordinates
CM is the covariance matrix of the residuals

Use this method to get eivenvalues and eigenvectors
{vects, vals, z} = SingularValueDecomposition[CM]

PCA is based on the covariance matrix

Diagonal elements are variances, off-diagonal are covariances (slopes)

	A	B	C
A	6.56	4.69	2.59
B	4.69	4.21	1.38
C	2.59	1.38	1.36

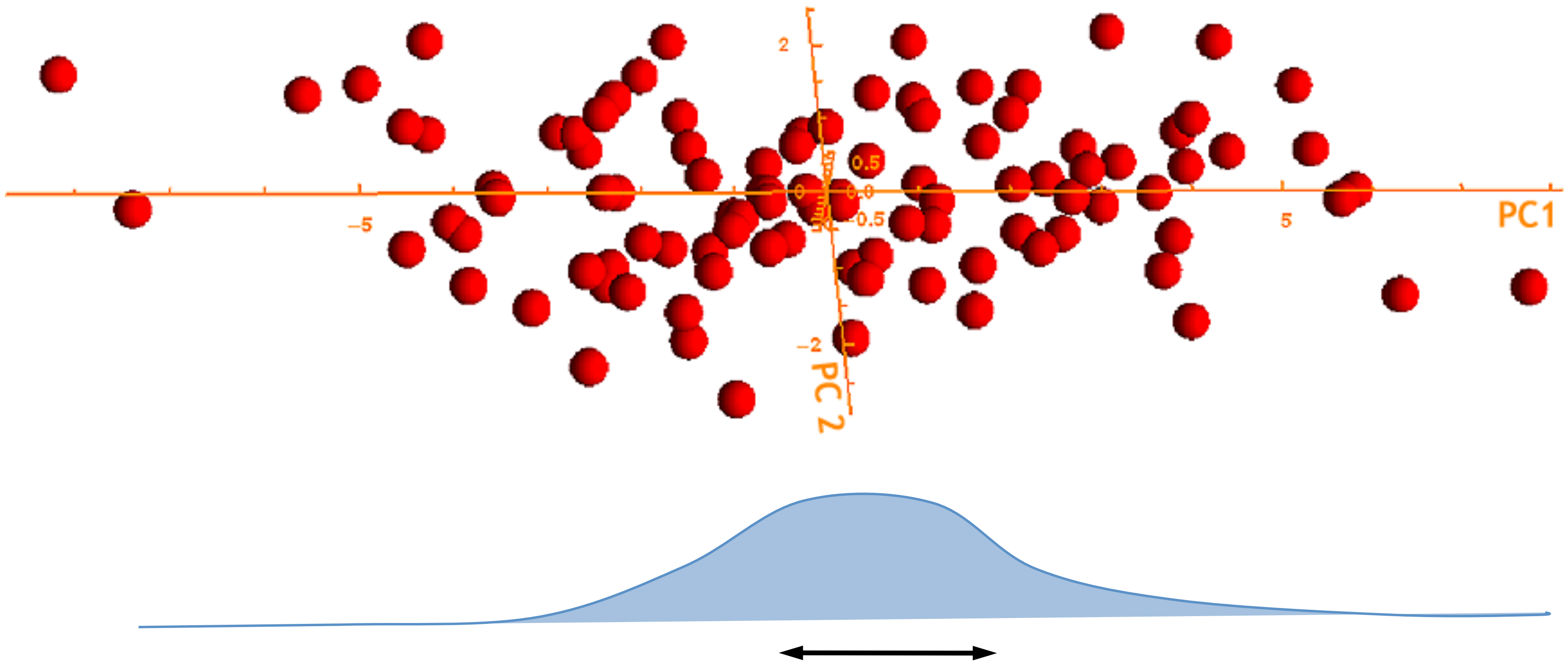


Eigenvalues

Variance of data along each PC axis

PC 1 = 11.08
PC 2 = 1.01
PC 3 = 0.04

	PC1	PC2	PC3
PC1	11.08	0	0
PC2	0	1.01	0
PC3	0	0	0.04



Important point: the meaning of eigenvalues

Between 95% and 99% of data lie within 2.0 SDs of the mean

1. If you know the variance, you know the standard deviation is its square-root;

2. You know that nearly all the data have a range of 4 * SD;

3. If the mean is 0.0, then nearly all the data lie between -2 * SD and +2 * SD;

4. The eigenvalues (or singular values) of a PC are variances, therefore the range of data on that PC can be calculated from them.

Important point: the meaning of eigenvalues (cont.)

Total variance of morphometric data set is the total amount of shape variation, which can be calculated three ways:

1. Summing squared distances between landmark points and the consensus (sample mean) for all the objects and dividing by (*n-1*);
2. Summing the eigenvalues that are returned by the PCA;
3. Summing squared PC scores (have a mean of zero so no subtraction is required) and dividing by (*n-1*);

If these three calculations don't give the same number, something is wrong

Useful variants on Eigenvalues

Eigenvalues

PC 1 = 11.08
PC 2 = 1.01
PC 3 = 0.04

12.13

Percent explained

100 * 11.08 / 12.13 = 91.3
100 * 1.01 / 12.13 = 8.3
100 * 0.04 / 12.13 = 0.3

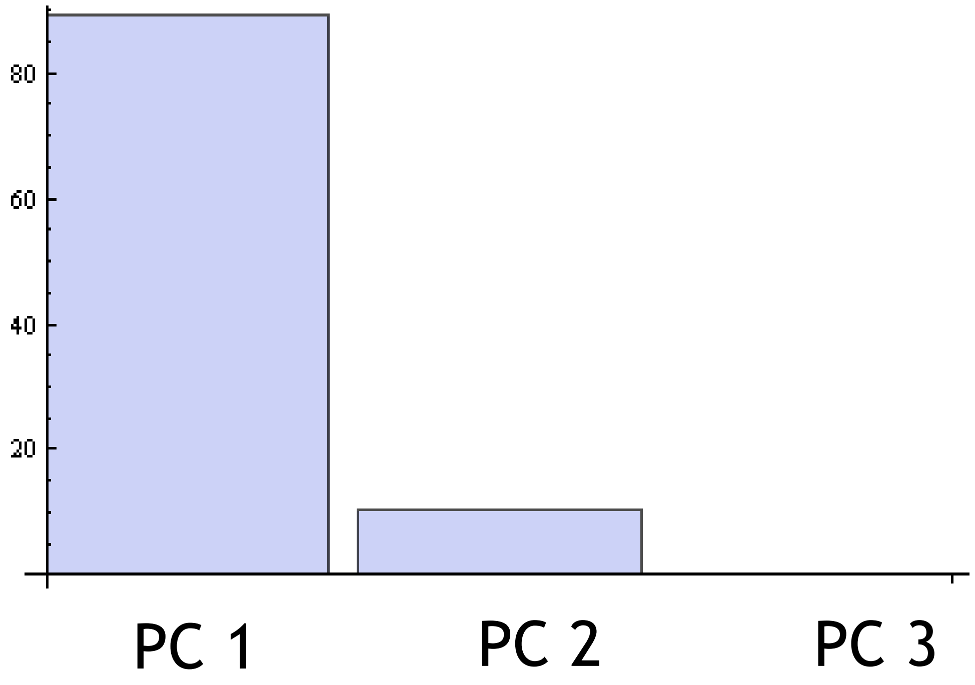
100.0

Standard Deviation

(procGPA reports this)

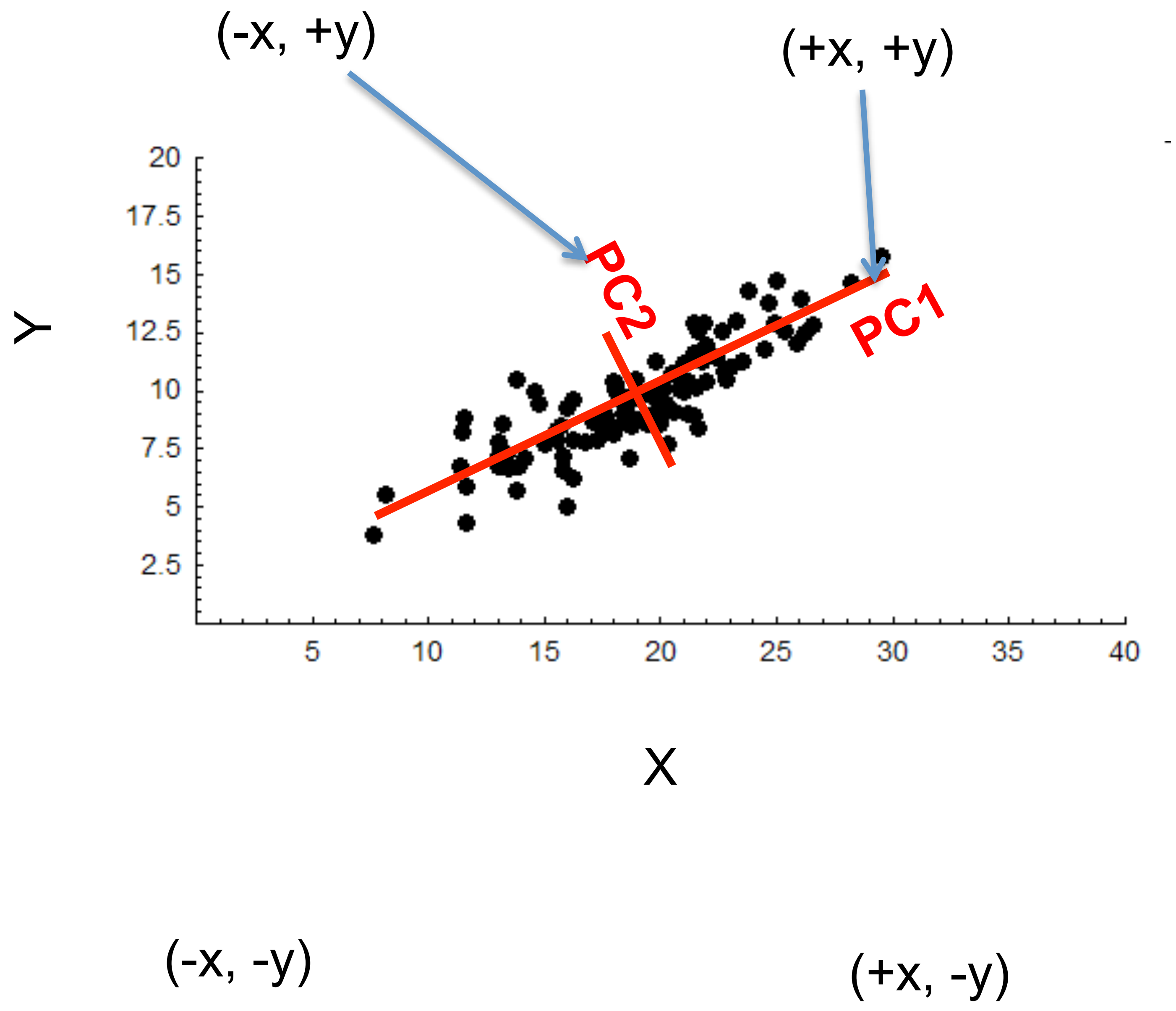
11.08^0.5 = 3.33
1.01^0.5 = 1.00
0.04^0.5 = 0.20

Scree plot



BarChart[Eigenvalues[CM]]

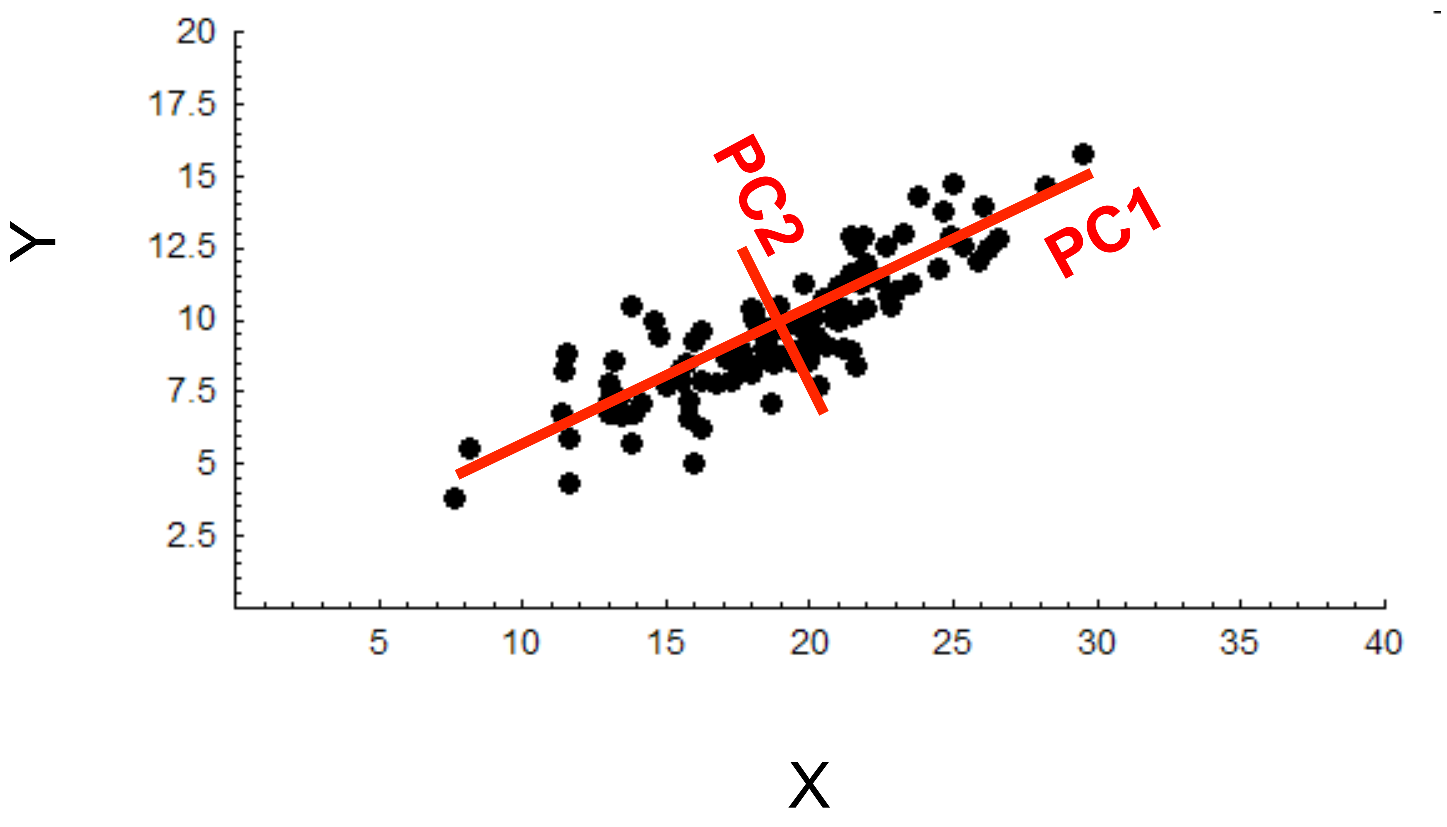
Eigenvector 'loadings' tell how each original variable contributes to the PC



Eigenvector Matrix

	PC1	PC2
X	0.89	-0.44
Y	0.44	0.89

Eigenvectors also describe how to transform data from original coordinate system to PCs and back



Eigenvector Matrix

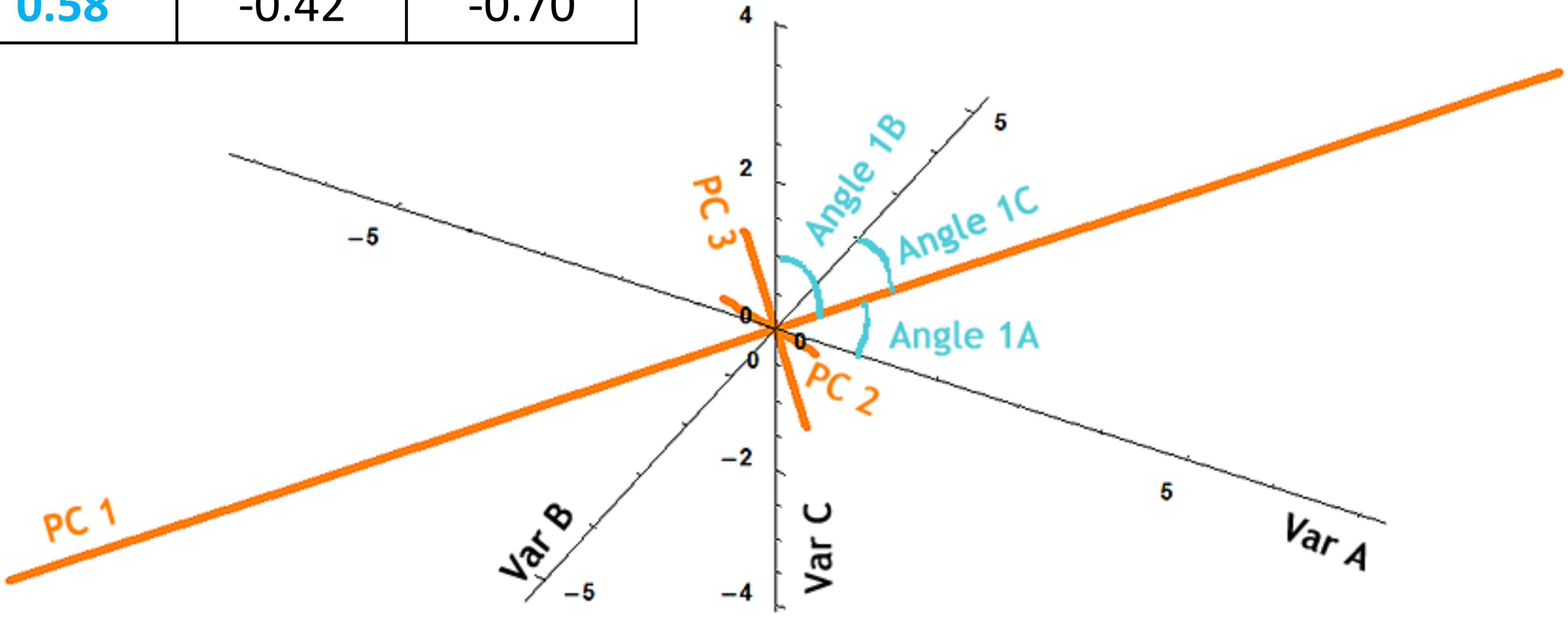
	PC1	PC2
X	0.89	-0.44
Y	0.44	0.89

(multiply PC1 X score by 0.89 and PC1 Y score by -0.44 and add back X,Y meanto get real X,Y)

Eigenvectors: definition 1

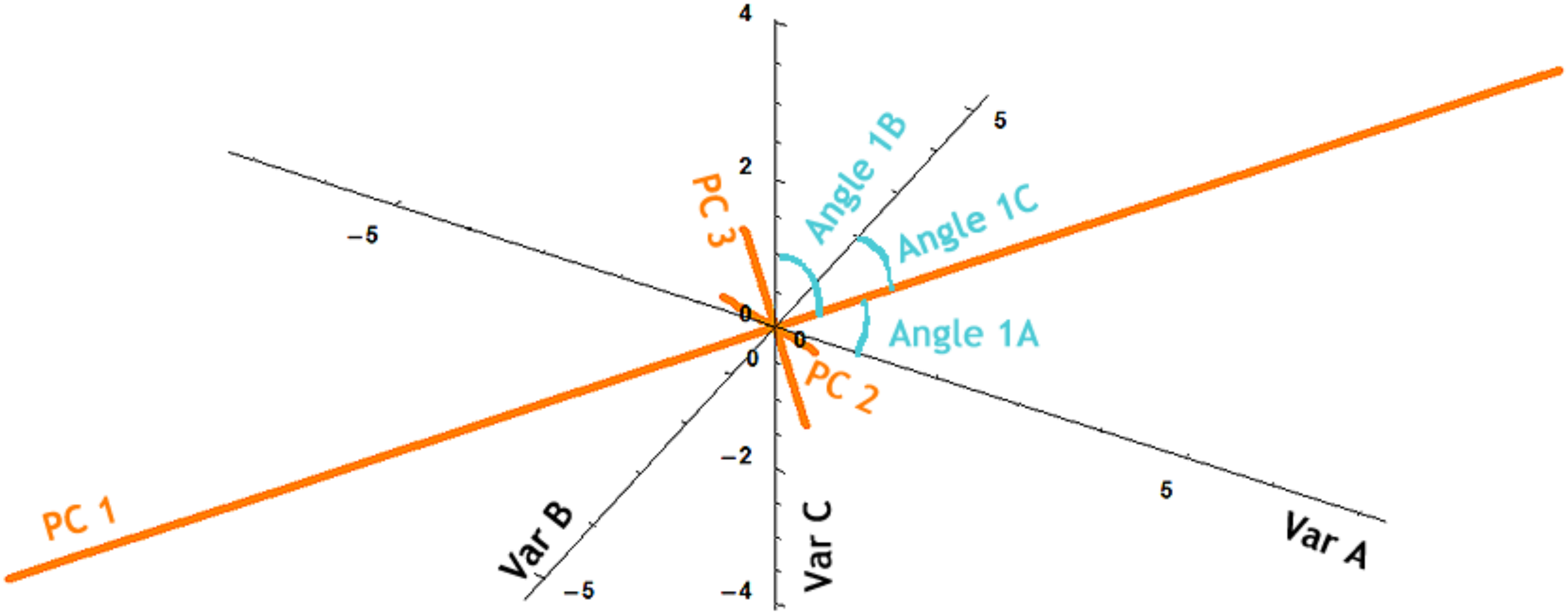
Angles between PC and original variables
(the eigen vector matrix is a rotation matrix in radians)

	PC1	PC2	PC3
Var A	-0.76	-0.58	-0.29
Var B	0.28	-0.69	0.66
Var C	0.58	-0.42	-0.70



Same Eigenvectors converted from radians to degrees

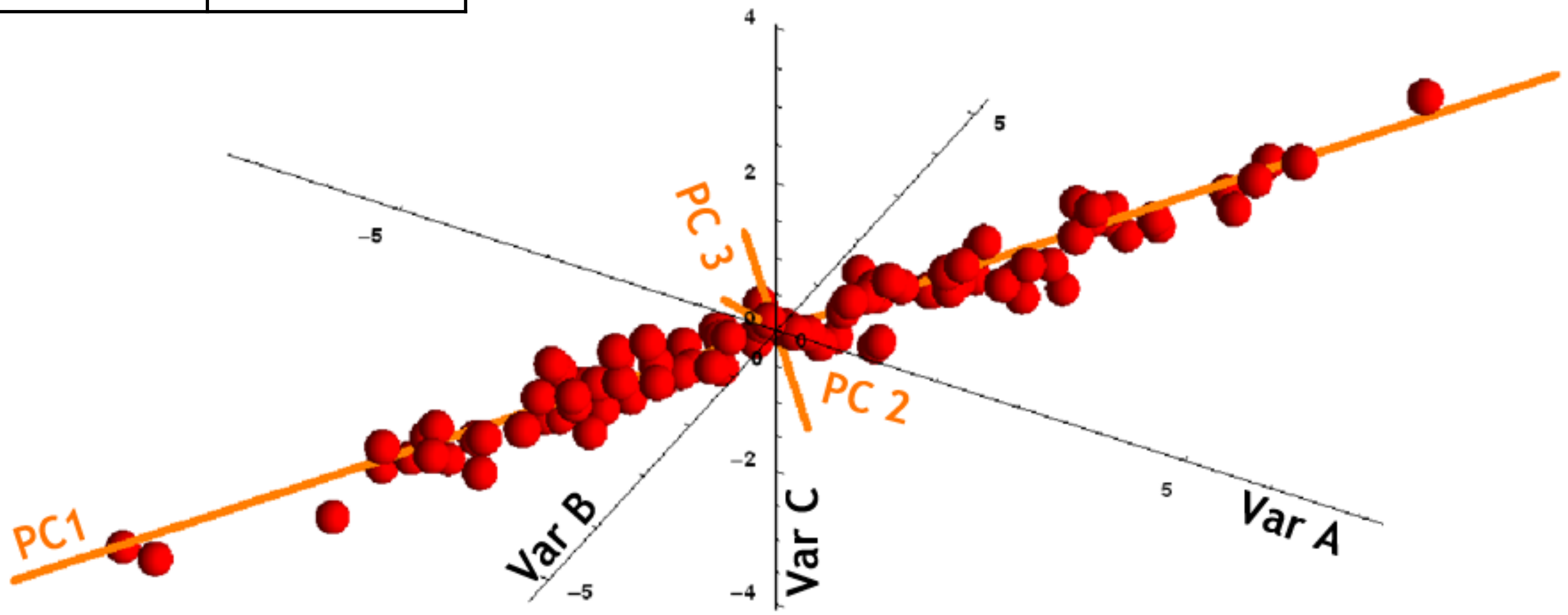
	PC1	PC2	PC3
Var A	-43.8	-33.1	-16.4
Var B	16.2	-39.9	37.7
Var C	33.2	-24.2	-39.9



Eigenvectors: definition 2

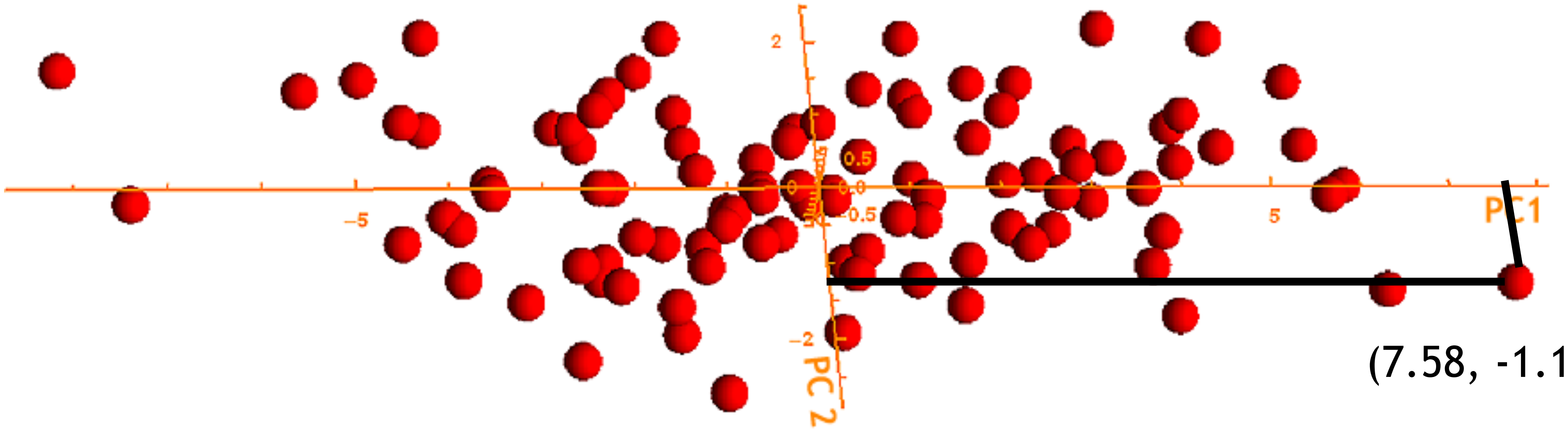
Loading (or importance) of each variable to the PC.
The larger the absolute value, the more important the variable.

	PC1	PC2	PC3
Var A	**-0.76**	-0.58	-0.29
Var B	**0.28**	-0.69	0.66
Var C	**0.58**	-0.42	-0.70



Scores

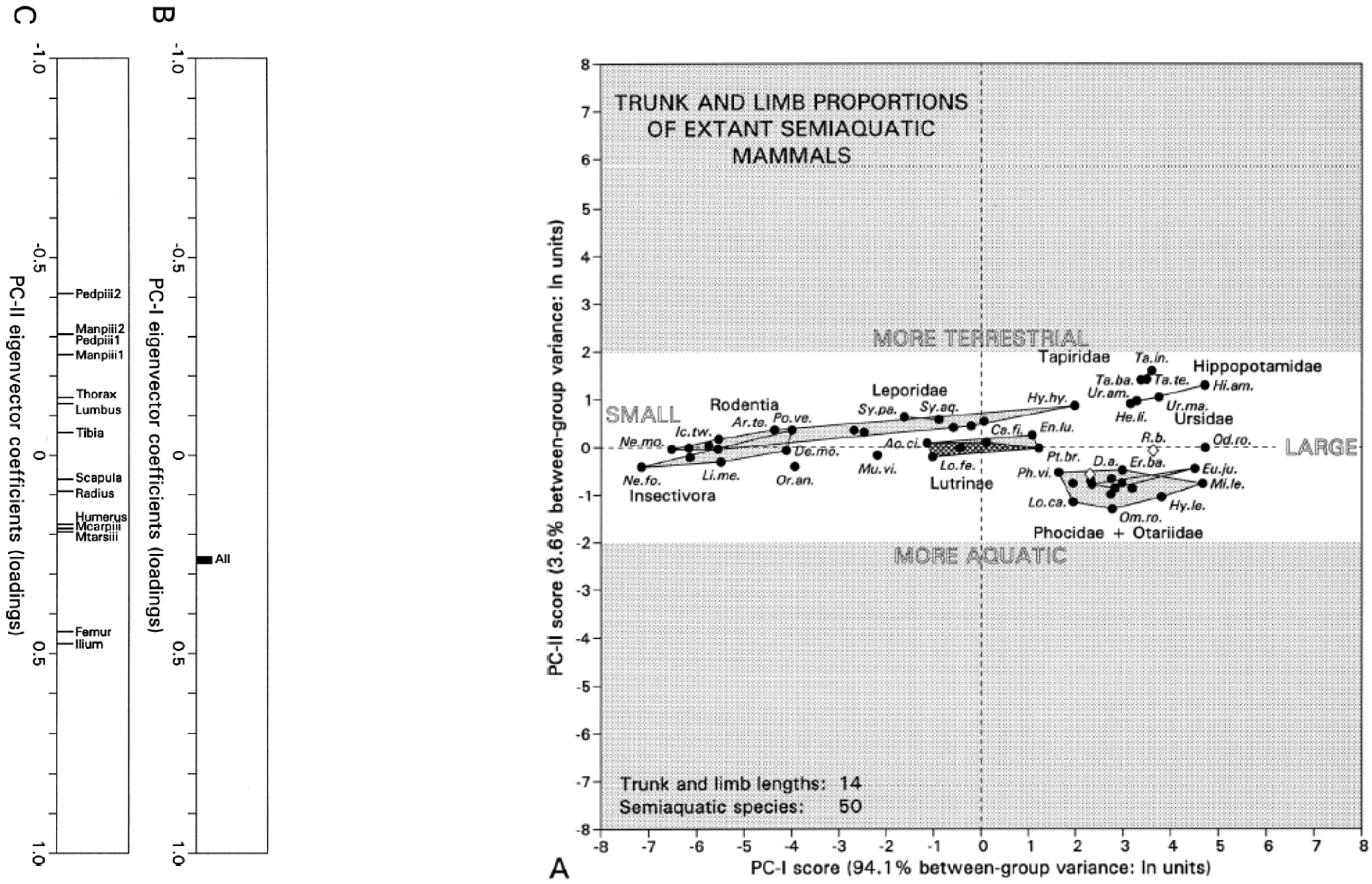
The coordinates of each data point on the PC axes. These numbers can be thought of as new variables, or shape variables.



(7.58, -1.18)

Score on PC1
and PC2

PCA plots have lots of meaning



Gingerich, P.D. 2003. Land-to-sea transition in early whales: evolution of Eocene archaeoceti (Cetacea) in relation to skeletal proportions and locomotion of living semiaquatic mammals. *Paleobiology*, 29: 429-454.

PCA is important in Geometric Morphometrics because....

1. PCA scores are used as shape variables
2. Eigenvectors are convenient axes for shape space
3. Eigenvectors and their scores are uncorrelated as variables
4. Variance (eigenvalues) is partitioned across eigenvectors and scores in descending order
5. Scores can be safely used for all other statistical analyses, including tree building
6. Eigenvectors can be used to build shape models

PCA vs Relative Warps vs Partial Warps

Relative warps = Principal components

Relative warps/Principal components organize shape variation so that the greatest amount is explained on PC1, second greatest on PC2, etc. Also PC1 is uncorrelated with PC2 is uncorrelated with PC3, etc.

Partial warps (can safely be ignored)

Partial Warps measure the “scale” of shape variation over the entire object down to a small part of the object. NOT principal components (even though the plots look alike). Partial warp 1 explains variation in ALL the landmarks, Partial warp 2 explains variation in part of the landmarks, Partial warp 3 in a smaller number, etc. Partial Warp 1 MAY be correlated with Partial warp 2, etc.

Shape modelling...

How to construct models of shapes in morphospace

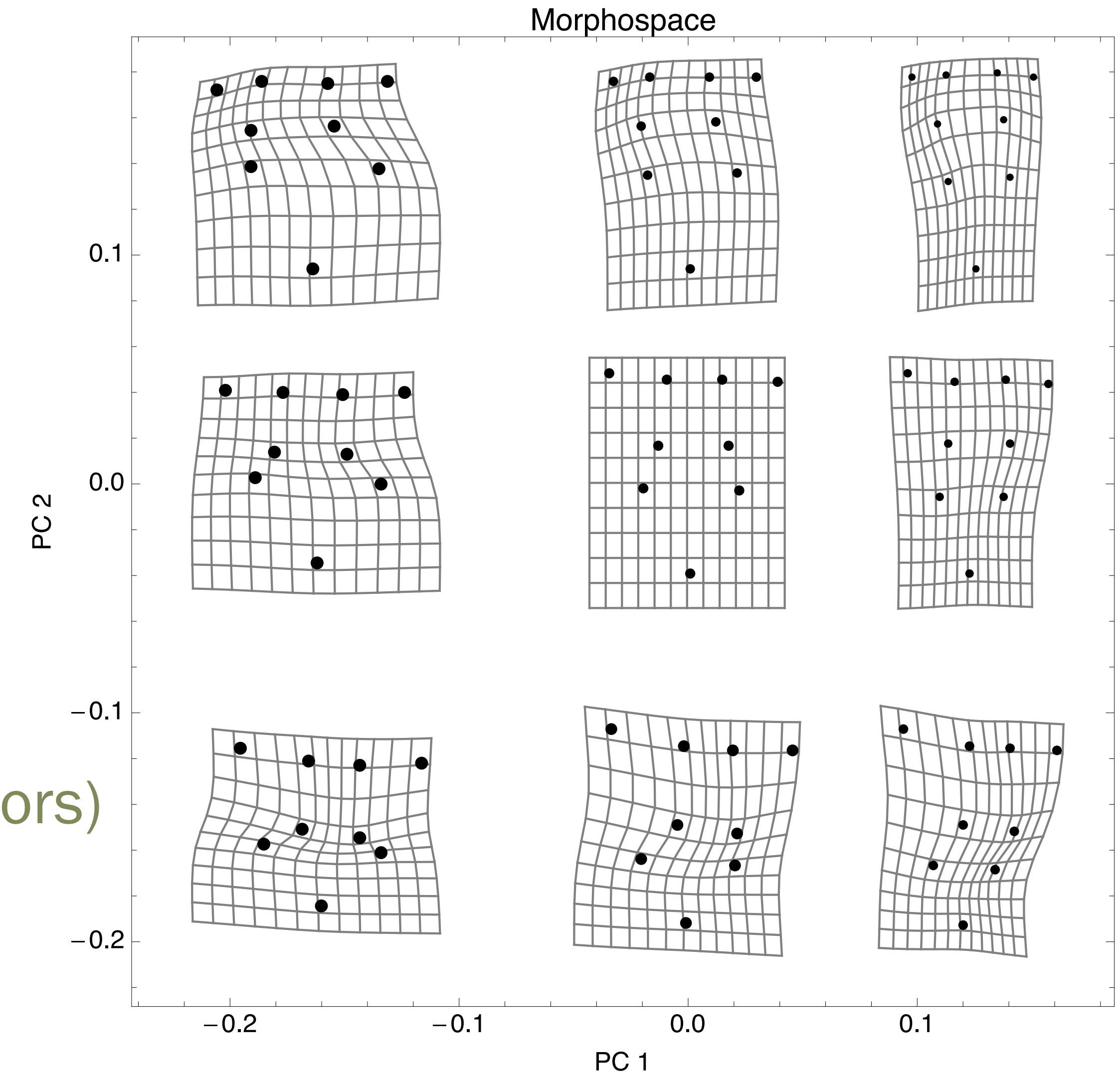
Ingredients:

1. mean shape (consensus)

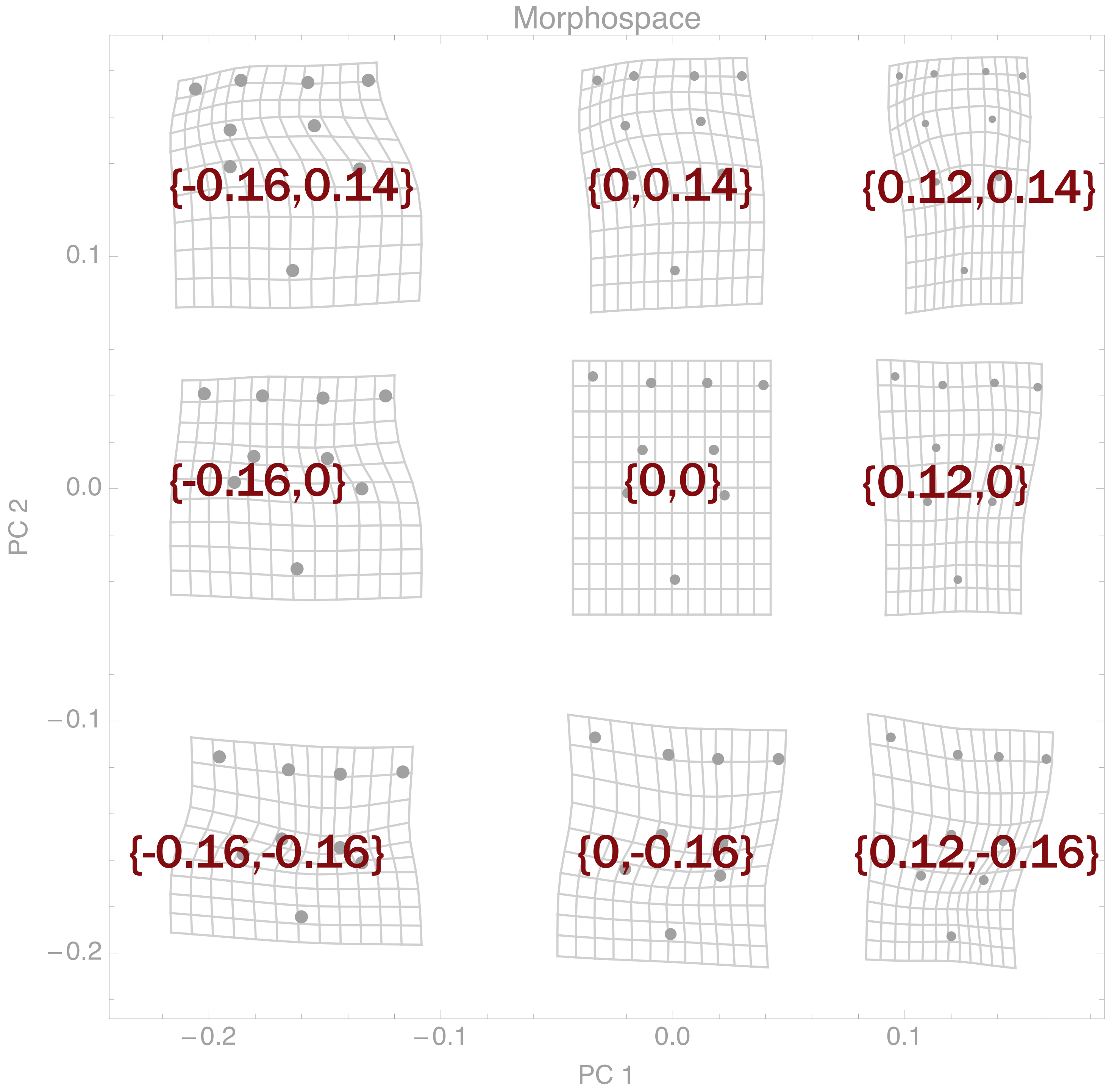
2. eigenvectors

3. the score (address) of the point to be modelled

Model = consensus + ∑ (scores * eigenvectors)



Shape modelling: note scores of the models



Shape models

Basic equation for 2D shape space:

model = score PC1 * vector PC1 + score PC2 * vector PC2 + consensus

Model at center of the space:

model = 0 * vector PC1 + 0 * vector PC2 + consensus
= consensus

Model at center right:

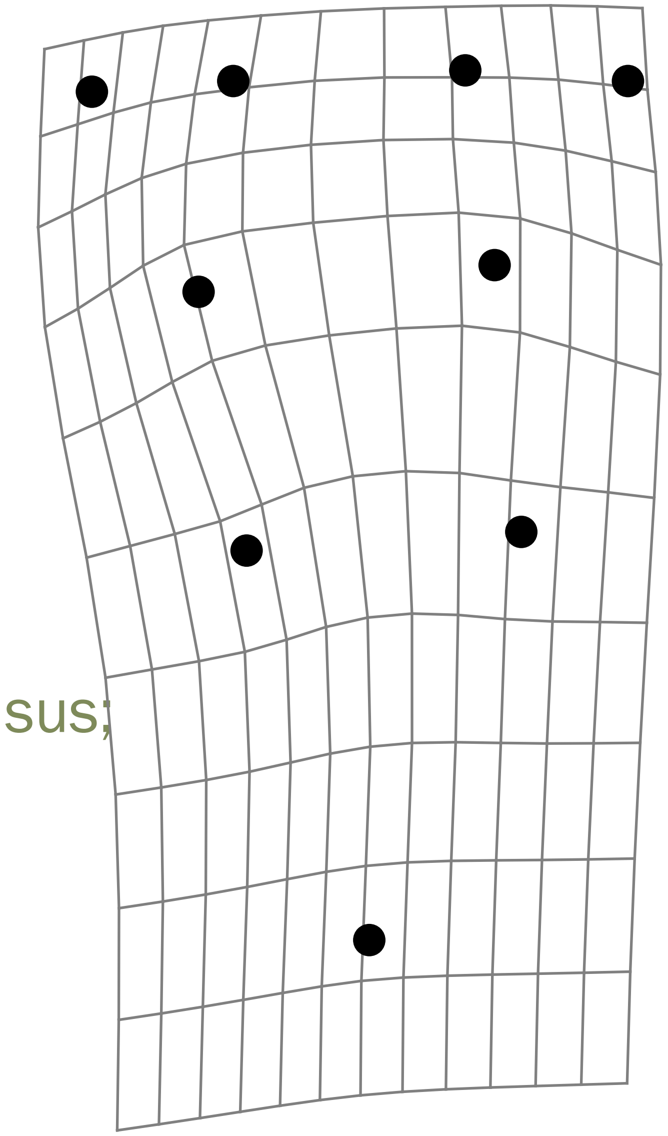
model = 0.12 * vector PC1 + 0 * vector PC2 + consensus
= 0.12 * vector PC1 + consensus

Model at upper right:

model = 0.12 * vector PC1 + 0.14 * vector PC2 + consensus

Shape models in Mathematica

```
proc = Procrustes[coords, 9, 2];

consensus = Mean[proc];

resids = Table[proc[[x]]-consensus, {x, Length[proc]}];

CM = Covariance[resids];

{vects, vals, z} = SingularValueDecomposition[CM];

model = 0.12 * vects[[1;;, 1]] + 0.14 * vects[[1;;, 2]] + consensus;

tpSpline[consensus, model]
```



Create your own modelling function

```
(* the next line sets up a user-defined function that takes three arguments, scores, vecs and consensus,
then does the calculation for the model *)
ShowModel[scores_, vecs_,consensus_]:= Plus@@(scores*vecs)+consensus
```

```
(* the next line uses the function defined in the previous line to create a model using the scores 0.12
and 0.14, the first two PC eigenvectors, and the consensus shape.  *)
ShowModel[{0.12,0.14}, vecs[[{1,2}]],consensus]
```

```
(* the next line repeats the model for a series of different PC1 scores, with PC2 held constant at 0 *)
Table[ShowModel[{x, 0}, vecs[[{1,2}]],consensus], {x, -0.12,0.12,0.05}]
```

```
(* the next line does the same as the last, but plugs the model into the tpSpline function showing how
the model differs from the mean shape. *)
Table[tpSpline[consensus, ShowModel[{x, 0}, vecs[[{1,2}]],consensus]], {x,
-0.12,0.12,0.05}]
```

```
(* the next line turns the last one into an animated movie that shows the range of shape from one end of
PC1 to the other. *)
ListAnimate[... same code as in previous line here... ]
```