Package ‘shapes’

February 3, 2023

Title  Statistical Shape Analysis
Date   2023-02-03
Version 1.2.7
Author Ian L. Dryden
Description Routines for the statistical analysis of landmark shapes, including Procrustes analysis, graphical displays, principal components analysis, permutation and bootstrap tests, thin-plate spline transformation grids and comparing covariance matrices.


Maintainer Ian Dryden <ian.dryden@nottingham.ac.uk>
Imports minpack.lm, scatterplot3d, rgl, MASS
Depends R (>= 2.10)
License GPL-2
URL http://www.maths.nottingham.ac.uk/~ild/shapes
NeedsCompilation no
Repository CRAN
Date/Publication 2023-02-03 21:50:01 UTC

R topics documented:

apes .......................................................... 3
backfit ......................................................... 3
bookstein2d .................................................. 4
brains .......................................................... 5
centroid.size ................................................ 6
cortical ......................................................... 7
digit3.dat .................................................... 8
distcov ......................................................... 8
dna.dat ......................................................... 9
estcov ......................................................... 10
<table>
<thead>
<tr>
<th>R topics documented:</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>frechet</td>
<td>11</td>
</tr>
<tr>
<td>gels</td>
<td>13</td>
</tr>
<tr>
<td>gorf.dat</td>
<td>13</td>
</tr>
<tr>
<td>gorm.dat</td>
<td>14</td>
</tr>
<tr>
<td>groupstack</td>
<td>15</td>
</tr>
<tr>
<td>humanmove</td>
<td>16</td>
</tr>
<tr>
<td>macaques</td>
<td>17</td>
</tr>
<tr>
<td>macf.dat</td>
<td>17</td>
</tr>
<tr>
<td>macm.dat</td>
<td>18</td>
</tr>
<tr>
<td>mice</td>
<td>19</td>
</tr>
<tr>
<td>panf.dat</td>
<td>20</td>
</tr>
<tr>
<td>panm.dat</td>
<td>20</td>
</tr>
<tr>
<td>plot3darcs</td>
<td>21</td>
</tr>
<tr>
<td>plotshapes</td>
<td>22</td>
</tr>
<tr>
<td>pns</td>
<td>23</td>
</tr>
<tr>
<td>pns4pc</td>
<td>25</td>
</tr>
<tr>
<td>pns33d</td>
<td>26</td>
</tr>
<tr>
<td>pongo.of.dat</td>
<td>27</td>
</tr>
<tr>
<td>pongom.dat</td>
<td>28</td>
</tr>
<tr>
<td>procdist</td>
<td>29</td>
</tr>
<tr>
<td>procGPA</td>
<td>30</td>
</tr>
<tr>
<td>procOPA</td>
<td>32</td>
</tr>
<tr>
<td>procWGPA</td>
<td>34</td>
</tr>
<tr>
<td>qce2.dat</td>
<td>36</td>
</tr>
<tr>
<td>qlet2.dat</td>
<td>37</td>
</tr>
<tr>
<td>qset2.dat</td>
<td>38</td>
</tr>
<tr>
<td>rats</td>
<td>38</td>
</tr>
<tr>
<td>resampletest</td>
<td>39</td>
</tr>
<tr>
<td>riemdist</td>
<td>41</td>
</tr>
<tr>
<td>rigidbody</td>
<td>42</td>
</tr>
<tr>
<td>sand</td>
<td>43</td>
</tr>
<tr>
<td>schizophrenia</td>
<td>44</td>
</tr>
<tr>
<td>schizophrenia.dat</td>
<td>45</td>
</tr>
<tr>
<td>shapemca</td>
<td>46</td>
</tr>
<tr>
<td>shapes.cva</td>
<td>47</td>
</tr>
<tr>
<td>shapes3d</td>
<td>48</td>
</tr>
<tr>
<td>shells</td>
<td>49</td>
</tr>
<tr>
<td>sooty</td>
<td>50</td>
</tr>
<tr>
<td>ssriemdists</td>
<td>51</td>
</tr>
<tr>
<td>steroids</td>
<td>52</td>
</tr>
<tr>
<td>testmeanshapes</td>
<td>53</td>
</tr>
<tr>
<td>tsgrid</td>
<td>55</td>
</tr>
<tr>
<td>transformations</td>
<td>56</td>
</tr>
</tbody>
</table>

Index 58
Great ape data

Description

Great ape skull landmark data. 8 landmarks in 2 dimensions, 167 individuals

Usage

data(apes)

Format

apes$x : An array of dimension 8 x 2 x 167
apes$group : Species and sex of each specimen: "gorf" 30 female gorillas, "gorm" 29 male gorillas, "panf" 26 female chimpanzees, "pamm" 28 male chimpanzees, "pongof" 24 female orang utans, "pongom" 30 male orang utans.

Source


References

Data from Paul O'Higgins (Hull-York Medical School)

Examples

data(apes)
par(mfrow=c(1,2))
plotshapes(apes$x[,apes$group=="gorf"],symbol="f")
plotshapes(apes$x[,apes$group=="gorm"],symbol="m")

backfit

Backfit from scores to configuration

Description

Backfit from PNSS or PCA scores to a representative configuration

Usage

backfit(scores, x, type="pnss", size=1)
Arguments

scores  n x p matrix of scores
x       An object that is the output of either pnss3d (if type="pnss") or procGPA (if type="pca")
type    Either "pnss" for PNSS or "pca" for PCA
size    The centroid size of the backfitted configuration. The default is 1 but one can rescale the backfitting if desired.

Value

A k x m matrix of co-ordinates of the backfitted configuration

Author(s)

Ian Dryden

References


See Also

pns, pns4pc, plot3darcs

Examples

ans <- pnss3d( macf.dat, sphere.type="BIC", n.pc=8)
y <- backfit( ans$PNS$scores[1,], ans ,type="pnss")
riemdist( macf.dat[,,1] , y ) #should be close to zero

ans2 <- procGPA( macf.dat, tangentcoords="partial")
y <- backfit( ans2$scores[1,] , ans2 ,type="pca")
riemdist( macf.dat[,,1], y ) #should be close to zero

---

bookstein2d  

*Bookstein’s baseline registration for 2D data*

Description

Carries out Bookstein’s baseline registration and calculates a mean shape

Usage

bookstein2d(A,l1=1,l2=2)
**brains**

**Description**

24 landmarks located in 58 adult healthy brains

**Usage**

data(brains)
Format

A list with components:

brains$x : An array of dimension 24 x 3 x 58 containing the landmarks in 3D
brains$sex : Sex of each volunteer (m or f)
brains$age : Age of each volunteer
brains$handed : Handedness of each volunteer (r or l)
brains$grp : Group label: 1= right-handed males, 2=left-handed males, 3=right-handed females,
4=left-handed females

References

(2001). Landmark-based morphometrics of the normal adult brain using MRI. Neuroimage, 13,
801–813.

Examples

data(brains)
# plot first three brains
shapes3d(brains$x[,,1:3])

---

**centroid.size**

**Centroid size**

Description

Calculate centroid size from a configuration or a sample of configurations.

Usage

centroid.size(x)

Arguments

x

For a single configuration k x m matrix or complex k-vector
For a sample of configurations k x m x n array or k x n complex matrix

Value

Centroid size(s)

Author(s)

Ian Dryden
References

Examples
```r
data(mice)
centroid.size(mice$x[,1])
```

---

description

Cortical surface data, from MR scans. Axial slice outlines with 500 points on each outline. 68 individuals.

Usage
```r
data(cortical)
```

Format
```r
cortical$age ( age) cortical$group ( Control, Schizophrenia) cortical$sex ( 1 = male, 2 = female) cortical$symm ( a symmetry measure from the original 3D cortical surface ) cortical$x (500 x , y coordinates of an axial slice through the cortical surface intersecting the anterior and posterior commissures) cortical$r (500 radii from equal angular polar coordinates )
```

Source

References
Original MR data from Sean Flynn (UBC) in collaboration with Bert Park (Nottingham).

Examples
```r
data(cortical)
plotshapes(cortical$x)
```
### digit3.dat

**Digit 3 data**

**Description**

Handwritten digit ‘3’ data. 13 landmarks in 2 dimensions, 30 individuals

**Usage**

```r
data(digit3.dat)
```

**Format**

An array of dimension 13 x 2 x 30

**Source**


**References**

http://www.maths.nott.ac.uk/personal/ild/bookdata/digit3.dat

Data from Cath Anderson

**Examples**

```r
data(digit3.dat)
k<-dim(digit3.dat)[1]
n<-dim(digit3.dat)[3]
plotshapes(digit3.dat, joinline=c(1:13))
```

---

### distcov

**Compute a distance between two covariance matrices**

**Description**

Compute a distance between two covariance matrices, with non-Euclidean options.

**Usage**

```r
distcov(S1, S2, method="Riemannian", alpha=1/2)
```
Arguments

S1 Input a covariance matrix (square, symmetric, positive definite)
S2 Input another covariance matrix of the same size
method The type of distance to be used: "Procrustes": Procrustes size-and-shape metric, "ProcrustesShape": Procrustes metric with scaling, "Riemannian": Riemannian metric, "Cholesky": Cholesky based distance, "Power": Power Euclidean, with power alpha, "Euclidean": Euclidean metric, "LogEuclidean": Log-Euclidean metric, "RiemannianLe": Another Riemannian metric.
alpha The power to be used in the power Euclidean metric

Value
The distance

Author(s)
Ian Dryden

References

See Also
estcov

Examples

```r
A <- diag(5)
B <- A + .1*matrix(rnorm(25),5,5)
S1<-A
S2<- B
distcov( S1, S2, method="Procrustes")
```

dna.dat DNA data

Description

Part of a 3D DNA molecule moving in time, k = 22 atoms, 30 time points
Usage

```r
data(dna.dat)
```

Format

An array of dimension 22 x 3 x 30

Examples

```r
data(dna.dat)
plotshapestime3d(dna.dat)
```

---

**estcov**

*Weighted Frechet mean of covariance matrices*

---

Description

Computes the weighted Frechet means of an array of covariance matrices, with different options for the covariance metric. Also carries out principal co-ordinate analysis of the covariance matrices.

Usage

```r
estcov(S, method="Riemannian", weights=1, alpha=1/2, MDSk=2)
```

Arguments

- **S**: Input an array of covariance matrices of size \( k \times k \times n \) where each matrix is square, symmetric and positive definite.
- **method**: The type of distance to be used: "Procrustes": Procrustes size-and-shape metric, "ProcrustesShape": Procrustes metric with scaling, "Riemannian": Riemannian metric, "Cholesky": Cholesky based distance, "Power": Power Euclidean, with power alpha, "Euclidean": Euclidean metric, "LogEuclidean": Log-Euclidean metric, "RiemannianLe": Another Riemannian metric.
- **weights**: The weights to be used for calculating the mean. If weights=1 then equal weights are used, otherwise the vector must be of length \( n \).
- **alpha**: The power to be used in the power Euclidean metric.
- **MDSk**: The number of MDS components in the principal co-ordinate analysis.

Value

A list with values

- **mean**: The weighted mean covariance matrix.
- **sd**: The weighted standard deviation.
- **pco**: Principal co-ordinates (from multidimensional scaling with the metric).
- **eig**: The eigenvalues from the principal co-ordinate analysis.
frechet

Author(s)
Ian Dryden

References

See Also
distcov

Examples
S <- array(0,c(5,5,10) )
for (i in 1:10){
  tem <- diag(5)+.1*matrix(rnorm(25),5,5)
  S[,,i]<- tem
}
estcov( S , method="Procrustes")

<table>
<thead>
<tr>
<th>frechet</th>
<th>Mean shape estimators</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description
Calculation of different types of Frechet mean shapes, or the isotropic offset Gaussian MLE mean shape

Usage
frechet(x, mean="intrinsic")

Arguments
x
Input k x m x n real array, where k is the number of points, m is the number of dimensions, and n is the sample size.

mean
Type of mean shape. The Frechet mean shape is obtained by minimizing sum d(x_i,nu)^2 with respect to nu. Different estimators are obtained with different choices of distance d. "intrinsic" intrinsic mean shape (d = rho = Riemannian distance); "partial.procrustes" partial Procrustes (d = 2*sin(rho/2)); "full.procrustes" full Procrustes (d = sin(rho)); h (positive real number) M-estimator (d^2 = (1 - cos^(2h)(rho))/h) Kent (1992); "mle" - isotropic offset Gaussian MLE of Mardia and Dryden (1989)
Value

A list with components

mshape  Mean shape estimate
var  Minimized Frechet variance (not available for MLE)
kappa  (if available) The estimated kappa for the MLE
code  Code from optimization, as given by function nlm - should be 1 or 2
gradient  Gradient from the optimization, as given by function nlm - should be close to zero

Author(s)

Ian Dryden

References


See Also

procGPA

Examples

#2D example : female and male Gorillas (cf. Dryden and Mardia, 2016)

data(gorf.dat)
frechet(gorf.dat[,1:4],mean="intrinsic")
**gels**  

*Electrophoresis gel data*

**Description**
Electrophoresis gel data. 10 invariant spots have been picked out by an expert on two electrophoretic gels.

**Usage**
```r
data(gels)
```

**Format**
An array of dimension 10 x 2 x 2

**Source**

**References**
Data from Chris Glasbey (BioSS)

**Examples**
```r
data(gels)
plotshapes(gels)
```

---

**gorf.dat**  

*Female gorilla data*

**Description**
Female gorilla skull data. 8 landmarks in 2 dimensions, 30 individuals

**Usage**
```r
data(gorf.dat)
```

**Format**
An array of dimension 8 x 2 x 30
Source


References

http://www.maths.nott.ac.uk/personal/ild/bookdata/gorf.dat
Data from Paul O’Higgins (Hull-York Medical School)

Examples

data(gorf.dat)
plotshapes(gorf.dat)

---

gorm.dat

Male gorilla data

Description

Male gorilla skull data. 8 landmarks in 2 dimensions, 29 individuals

Usage

data(gorm.dat)

Format

An array of dimension 8 x 2 x 29

Source


References

http://www.maths.nott.ac.uk/personal/ild/bookdata/gorm.dat
Data from Paul O’Higgins (Hull-York Medical School)

Examples

data(gorm.dat)
plotshapes(gorm.dat)
groupstack

Combine two or more groups of configurations

Description

Combine two or more groups of configurations and create a group label vector. (Maximum 8 groups).

Usage

groupstack(A1, A2, A3=0, A4=0, A5=0, A6=0, A7=0, A8=0)

Arguments

A1 Input \( k \times m \times n \) real array of the Procrustes transformed configurations, where \( k \) is the number of points, \( m \) is the number of dimensions, and \( n \) is the sample size.

A2 Input \( k \times m \times n \) real array of the Procrustes original configurations, where \( k \) is the number of points, \( m \) is the number of dimensions, and \( n \) is the sample size.

A3 Optional array

A4 Optional array

A5 Optional array

A6 Optional array

A7 Optional array

A8 Optional array

Value

A list with components

\( x \) The combined array of all configurations

\( \text{groups} \) The group labels (integers)

Author(s)

Ian Dryden

References


See Also

procGPA
Examples

# 2D example: female and male Gorillas (cf. Dryden and Mardia, 2016)

data(gorf.dat)
data(gorm.dat)
groupstack(gorf.dat,gorm.dat)

humanmove Human movement data

Description
Human movement data. 4 landmarks in 2 dimensions, 5 individuals observed at 10 times.

Usage
data(humanmove)

Format
humanmove: An array of landmark configurations 4 x 2 x 10 x 5

Source

References
Data from James Richardson.

Examples
data(humanmove)
# plotshapes(humanmove[,1])
# for (i in 2:5){
# for (j in 1:4){
# for (k in 1:10){
# points(humanmove[j,k,i],col=i)
#
#}
#}
#}
Description

Male and female macaque skull data. 7 landmarks in 3 dimensions, 18 individuals (9 males, 9 females)

Usage

data(macaques)

Format

macaques$x : An array of dimension 7 x 3 x 18
macaques$group : A factor indicating the sex (‘m’ for male and ‘f’ for female)

Source


References


Examples

data(macaques)
shapes3d(macaques$x[,1])

Description

Female macaque skull data. 7 landmarks in 3 dimensions, 9 individuals

Usage

data(macf.dat)

Format

An array of dimension 7 x 3 x 9
Source


References

Data from Paul O’Higgins (Hull-York Medical School)

Examples

data(macf.dat)
plotshapes(macf.dat)

macm.dat
Male macaque data

Description

Male macaque skull data. 7 landmarks in 3 dimensions, 9 individuals

Usage

data(macm.dat)

Format

An array of dimension 7 x 3 x 9

Source


References

Data from Paul O’Higgins (Hull-York Medical School)

Examples

data(macm.dat)
plotshapes(macm.dat)
**mice**

*T2 mouse vertabrae data*

Description

T2 mouse vertebrae data - 6 landmarks in 2 dimensions, in 3 groups (30 Control, 23 Large, 23 Small mice). The 6 landmarks are obtained using a semi-automatic method at points of high curvature. This particular strain of mice is the ‘QE’ strain. In addition pseudo-landmarks are given around each outlines.

Usage

data(mice)

Format

mice$x : An array of dimension 6 x 2 x 76 of the two dimensional co-ordinates of 6 landmarks for each of the 76 mice.

mice$group : Group labels. "c" Control, "l" Large, "s" Small mice

mice$outlines : An array of dimension 60 x 2 x 76 containing the 6 landmarks and 54 pseudo-landmarks, with 9 pseudo-landmarks approximately equally spaced between each pair of landmarks.

Source


References


Data from Paul O’Higgins (Hull-York Medical School) and David Johnson (Leeds)

Examples

data(mice)

plotshapes(mice$x,symbol=as.character(mice$group),joinline=c(1,6,2:5,1))
**panf.dat**  
*Female chimpanzee data*

**Description**
Female chimpanzee skull data. 8 landmarks in 2 dimensions, 26 individuals

**Usage**

```r
data(panf.dat)
```

**Format**
An array of dimension 8 x 2 x 26

**Source**

**References**
Data from Paul O’Higgins (Hull-York Medical School)

**Examples**

```r
data(panf.dat)
plotshapes(panf.dat)
```

---

**panm.dat**  
*Male chimpanzee data*

**Description**
Male chimpanzee skull data. 8 landmarks in 2 dimensions, 28 individuals

**Usage**

```r
data(panm.dat)
```

**Format**
An array of dimension 8 x 2 x 28
plot3darcs

Source


References

Data from Paul O’Higgins (Hull-York Medical School)

Examples

data(panm.dat)
plotshapes(panm.dat)

plot3darcs Modes of variation plots for PCA and PNSS

Description

Modes of variation plots for PCA and PNSS based on 3D views and arcs along a mode. \( c \times \text{sd} \) : the extent along lower and upper principal arcs. The lower principal arc \( \rightarrow 0 \rightarrow \) upper principal arc has a total of \( 2\times nn+1 \) configurations with: \( nn \) configurations along the negative principal arc to 0; one configuration at the PNS mean; \( nn \) configurations along the positive principal arc.

Usage

plot3darcs(x, pcno=1, c=1, nn=100, boundary.data=TRUE, view.theta=0, view.phi=0, type="pnss")

Arguments

- **x**: Output from pnss3d
- **pcno**: The number of the PC/PNSS component. The default is 1, the first PC/PNSS
- **c**: Number of standard deviations along each arc
- **nn**: In total \( 2 \times nn + 1 \) configurations: \( n \) configurations on arc from negative to 0; 1 configuration at 0; \( nn \) configurations from 0 to positive
- **boundary.data**: Logical for whether to use boundary data or not.
- **view.theta**: Viewing angle theta
- **view.phi**: Viewing angle phi
- **type**: "pnss" principal nested sphere mean and arc, or "pca" Procrustes mean and linear PC.

Value

A list with components

- **PNSmean**: the PNSS mean
- **lu.arc**: the configurations along the arc
Author(s)

Kwang-Rae Kim, Ian Dryden

References


See Also

pns, pns4pc, pnss3d

Examples

```r
ans <- pnss3d(digit3.dat, sphere.type="BIC", n.pc=5)
#aa <- plot3darcs(ans,c=2,pcno=1)
#bb <- plot3darcs(ans,c=2,pcno=1,type="pca")
```

Description

Plots configurations. Either one or two groups of observations can be plotted on the same scale.

Usage

```r
plotshapes(A, B = 0, joinline = c(1, 1), orthproj=c(1,2),color=1,symbol=1)
```

Arguments

- `A` : A k x m x n array, or k x m matrix for first group
- `B` : A k x m x n array, or k x m matrix for 2nd group (can be missing)
- `joinline` : A vector stating which landmarks are joined up by lines, e.g. joinline=c(1:n,1) will start at landmark 1, join to 2, ..., join to n, then re-join to landmark 1.
- `orthproj` : A vector stating which two orthogonal projections will be used. For example, for m=3 dimensional data: X-Y projection given by c(1,2) (default), X-Z projection given by c(1,3), Y-Z projection given by c(2,3).
- `color` : Colours for points. Can be a vector, e.g. 1:k gives each landmark a different colour for the specimens
- `symbol` : Plotting symbols. Can be a vector, e.g. 1:k gives each landmark a different symbol for the specimens
Principal Nested Spheres

Description

Calculation of Principal Nested Spheres

Usage

```r
pns(x, sphere.type = "seq.test", alpha = 0.1, R = 100,
    nlast.small.sphere = 1, output=TRUE)
```

Arguments

- `x` a \((d + 1) \times n\) data matrix where each column is a unit vector in \(S^d\) and \(n\) is the sample size.
- `sphere.type` a character string specifying the type of sphere fitting method. "seq.test" specifies sequential tests to decide either "small" or "great"; "small" specifies Principal Nested SMALL Sphere; "great" specifies Principal Nested GREAT Sphere (radius \(\pi/2\)); "BIC" specifies BIC statistic to decide either "small" or "great"; and "bi.sphere" specifies Principal Nested GREAT Sphere for the first part and Principal Nested SMALL Sphere for last parts. The default is "seq.test".
- `alpha` significance level \((0 < \alpha < 1)\) used when sphere.type = "seq.test". The default is 0.1.
- `R` the number of bootstrap samples to be evaluated for the sequential test. The default is 100.
nlast.smallsphere
the number of small spheres in the finishing part used when sphere.type = "bi.sphere".

output Logical. If TRUE then plots and some brief printed summaries are given. If FALSE then no plots or output is given.

Value
A list with components

resmat the residual matrix (X_PNS). Each entry in row k works like the kth principal component
$PNS = the list with the following components.
radii the size (radius) of PNS.
ortaxis the orthogonal axis v_i of subspheres.
dist the distance r_i of subspheres
pvalues the p-values of LRT and parametric bootstrap tests (if any).
ratio the estimated ratios. Now unavailable.
mean the location of the PNS mean.
sphere.type the type of method for fitting subspheres.
percent proportion of variances explained.
spherePNS The co-ordinates of the data points projected to the sphere in 3D (also plotted)
circlePNS The co-ordinates of the 2D circle projections on the sphere in 3D (also plotted)

Author(s)
Kwang-Rae Kim: R translation of Sungkyu Jung’s matlab code

References

See Also
pns4pc, pnss3d

Examples

```r
# out <- pc2sphere(x = gorf.dat, n.pc = 2)
# spheredata <- t(out$spheredata)
# pns.out <- pns(x = spheredata)
```
pns4pc

Principal Nested Shape Spaces from PCA

Description

Approximation of Principal Nested Shapes Spaces using PCA

Usage

pns4pc(x, sphere.type = "seq.test", alpha = 0.1, R = 100, nlast.small.sphere = 1, n.pc=2)

Arguments

- **x**: k x m x n array of landmark data.
- **sphere.type**: a character string specifying the type of sphere fitting method. "seq.test" specifies sequential tests to decide either "small" or "great"; "small" specifies Principal Nested SMALL Sphere; "great" specifies Principal Nested GREAT Sphere (radius π/2); "BIC" specifies BIC statistic to decide either "small" or "great"; and "bi.sphere" specifies Principal Nested GREAT Sphere for the first part and Principal Nested SMALL Sphere for The default is "seq.test".
- **alpha**: significance level (0 < alpha < 1) used when sphere.type = "seq.test". The default is 0.1.
- **R**: the number of bootstrap samples to be evaluated for the sequential test. The default is 100.
- **nlast.small.sphere**: the number of small spheres in the finishing part used when sphere.type = "bi.sphere".
- **n.pc**: the number of PC scores to be used (n.pc >= 2)

Value

A list with components

- **PNS**: the output of the function pns
- **GPAout**: the result of GPA
- **spheredata**: transformed spherical data from the PC scores
- **percent**: proportion of variances explained.

Author(s)

Kwang-Rae Kim

References


pnss3d

Principal Nested Shape Space Analysis

Description

Approximation of Principal Nested Shapes Spaces using PCA: 2D or 3D data, small or large samples

Usage

```
pnss3d(x, sphere.type="seq.test", alpha = 0.1, R = 100,
       nlast.small.sphere = 1, n.pc="Full", output=TRUE)
```

Arguments

- **x**: k x m x n array of landmark data.
- **sphere.type**: a character string specifying the type of sphere fitting method. "seq.test" specifies sequential tests to decide either "small" or "great"; "small" specifies Principal Nested SMALL Sphere; "great" specifies Principal Nested GREAT Sphere (radius pi/2); "BIC" specifies BIC statistic to decide either "small" or "great"; and "bi.sphere" specifies Principal Nested GREAT Sphere for the first part and Principal Nested SMALL Sphere for the last part. The default is "seq.test".
- **alpha**: significance level (0 < alpha < 1) used when sphere.type = "seq.test". The default is 0.1.
- **R**: the number of bootstrap samples to be evaluated for the sequential test. The default is 100.
- **nlast.small.sphere**: the number of small spheres in the finishing part used when sphere.type = "bi.sphere".
- **n.pc**: the number of PC scores to be used (n.pc >= 2)
- **output**: Logical. If TRUE then plots and some brief printed summaries are given. If FALSE then no plots or output is given.

See Also

pns, pns4pc, pnss3d, plot3darcs

Examples

```
pns4pc(digit3.dat, n.pc=2)
```
pongof.dat

Value

A list with components

PNS          the output of the function pns
GPAout       the result of GPA
spheredata   transformed spherical data from the PC scores
percent      proportion of variances explained.

Author(s)

Kwang-Rae Kim, Ian Dryden

References


See Also

pns, pns4pc, plot3darcs

Examples

ans <- pnss3d(digit3.dat, sphere.type="BIC", n.pc=5)
#aa <- plot3darcs(ans,c=2,pcno=1)
#bb <- plot3darcs(ans,c=2,pcno=1,type="pca")

pongof.dat

Female orang utan data

Description

Female orang utan skull data. 8 landmarks in 2 dimensions, 30 individuals

Usage

data(pongof.dat)

Format

An array of dimension 8 x 2 x 30

Source

References

Data from Paul O’Higgins (Hull-York Medical School)

Examples

data(pongof.dat)
plotshapes(pongof.dat)

---

pongom.dat  Male orang utan data

Description

Male orang utan skull data. 8 landmarks in 2 dimensions, 30 individuals

Usage

data(pongom.dat)

Format

An array of dimension 8 x 2 x 30

Source


References

Data from Paul O’Higgins (Hull-York Medical School)

Examples

data(pongom.dat)
plotshapes(pongom.dat)
**procdist**

*Procrustes distance*

**Description**
Calculates different types of Procrustes shape or size-and-shape distance between two configurations.

**Usage**

```r
procdist(x, y, type="full", reflect=FALSE)
```

**Arguments**

- `x`: k x m matrix (or complex k-vector for 2D data) where k = number of landmarks and m = no of dimensions
- `y`: k x m matrix (or complex k-vector for 2D data)
- `type`: string indicating the type of distance; "full" full Procrustes distance, "partial" partial Procrustes distance, "Riemannian" Riemannian shape distance, "sizeandshape" size-and-shape Riemannian/Procrustes distance
- `reflect`: Logical. If reflect = TRUE then reflection invariance is included.

**Value**
The distance between the two configurations.

**Author(s)**
Ian Dryden

**References**

**See Also**
procOPA, procGPA

**Examples**

```r
data(gorf.dat)
data(gorm.dat)
gorf<-procGPA(gorf.dat)
gorm<-procGPA(gorm.dat)
distfull<-procdist(gorf$mshape,gorm$mshape)
cat("Full Procrustes distance between mean shapes is ",distfull," \n")
```
Generalised Procrustes analysis

Description

Generalised Procrustes analysis to register landmark configurations into optimal registration using translation, rotation and scaling. Reflection invariance can also be chosen, and registration without scaling is also an option. Also, obtains principal components, and some summary statistics.

Usage

procGPA(x, scale = TRUE, reflect = FALSE, eigen2d = FALSE, tol1 = 1e-05, tol2 = tol1, tangentcoords = "residual", proc.output=FALSE, distances=TRUE, pcaoutput=TRUE, alpha=0, affine=FALSE)

Arguments

x Input k x m x n real array, (or k x n complex matrix for m=2 is OK), where k is the number of points, m is the number of dimensions, and n is the sample size.
scale Logical quantity indicating if scaling is required
reflect Logical quantity indicating if reflection is required
eigen2d Logical quantity indicating if complex eigenanalysis should be used to calculate Procrustes mean for the particular 2D case when scale=TRUE, reflect=FALSE
tol1 Tolerance for optimal rotation for the iterative algorithm: tolerance on the mean sum of squares (divided by size of mean squared) between successive iterations
tol2 tolerance for rescale/rotation step for the iterative algorithm: tolerance on the mean sum of squares (divided by size of mean squared) between successive iterations
tangentcoords Type of tangent coordinates. If (SCALE=TRUE) the options are "residual" (Procrustes residuals, which are approximate tangent coordinates to shape space), "partial" (Kent’s partial tangent co-ordinates), "expomap" (tangent coordinates from the inverse of the exponential map, which are the similar to "partial" but scaled by (rho/sin(rho)) where rho is the Riemannian distance to the pole of the projection. If (SCALE=FALSE) then all three options give the same tangent co-ordinates to size-and-shape space, which is simply the Procrustes residual X^P - mu.
proc.output Logical quantity indicating if printed output during the iterations of the Procrustes GPA algorithm should be given
distances Logical quantity indicating if shape distances and sizes should be calculated
cpoutput Logical quantity indicating if PCA should be carried out
alpha The parameter alpha used for relative warps analysis, where alpha is the power of the bending energy matrix. If alpha = 0 then standard Procrustes PCA is carried out. If alpha = 1 then large scale variations are emphasized, if alpha = -1 then small scale variations are emphasised. Requires m=2 and m=3 dimensional data if alpha != 0.
**affine** Logical. If TRUE then only the affine subspace of shape variability is considered.

**Value**

A list with components

- **k** no of landmarks
- **m** no of dimensions (m-D dimension configurations)
- **n** sample size
- **mshape** Procrustes mean shape. Note this is unit size if complex eigenanalysis used, but on the scale of the data if iterative GPA is used.
- **tan** The tangent shape (or size-and-shape) coordinates
- **rotated** the k x m x n array of full Procrustes rotated data
- **pcar** the columns are eigenvectors (PCs) of the sample covariance $S_v$ of tan
- **pcasd** the square roots of eigenvalues of $S_v$ using tan (s.d.’s of PCs)
- **percent** the percentage of variability explained by the PCs using tan. If alpha $\neq 0$ then it is the percent of non-affine variation of the relative warp scores. If affine is TRUE it is the percentage of total shape variability of each affine component.
- **size** the centroid sizes of the configurations
- **stdscores** standardised PC scores (each with unit variance) using tan
- **rawscores** raw PC scores using tan
- **rho** Kendall’s Riemannian shape distance rho to the mean shape
- **rmsrho** root mean square (r.m.s.) of rho
- **rmsd1** r.m.s. of full Procrustes distances to the mean shape $d_F$
- **GSS** Minimized Procrustes sum of squares

**Author(s)**

Ian Dryden, with input from Mohammad Faghihi and Alfred Kume

**References**


See Also

procOPA, riemdist, shapepca, testmeanshapes

Examples

# 2D example: female and male Gorillas (cf. Dryden and Mardia, 2016)

data(gorf.dat)
data(gorm.dat)

plotshapes(gorf.dat, gorm.dat)
n1 <- dim(gorf.dat)[3]
n2 <- dim(gorm.dat)[3]
k <- dim(gorf.dat)[1]
m <- dim(gorf.dat)[2]
gor.dat <- array(0, c(k, 2, n1 + n2))
gor.dat[, 1:n1] <- gorf.dat
gor.dat[, (n1 + 1):(n1 + n2)] <- gorm.dat

gor <- procGPA(gor.dat)
shapepca(gor, type = "r", mag = 3)
shapepca(gor, type = "v", mag = 3)

gor.gp <- c(rep("f", times = 30), rep("m", times = 29))
x <- cbind(gor$size, gor$rho, gor$scores[, 1:3])
pairs(x, panel = function(x, y) text(x, y, gor.gp),
      label = c("s", "rho", "score 1", "score 2", "score 3"))

# 3D example

data(macm.dat)
out <- procGPA(macm.dat, scale = FALSE)

par(mfrow = c(2, 2))

plot(out$rawscores[, 1], out$rawscores[, 2], xlab = "PC1", ylab = "PC2")
title("PC scores")
plot(out$rawscores[, 2], out$rawscores[, 3], xlab = "PC2", ylab = "PC3")
plot(out$rawscores[, 1], out$rawscores[, 3], xlab = "PC1", ylab = "PC3")
plot(out$size, out$rho, xlab = "size", ylab = "rho")
title("Size versus shape distance")

procOPA

Ordinary Procrustes analysis
**procOPA**

**Description**

Ordinary Procrustes analysis: the matching of one configuration to another using translation, rotation and (possibly) scale. Reflections can also be included if desired. The function matches configuration B onto A by least squares.

**Usage**

```r
procOPA(A, B, scale = TRUE, reflect = FALSE)
```

**Arguments**

- **A**
  - A k x m matrix (or complex k-vector for 2D data), of k landmarks in m dimensions. This is the reference figure.

- **B**
  - A k x m matrix (or complex k-vector for 2D data). This is the figure which is to be transformed.

- **scale**
  - logical indicating if scaling is required

- **reflect**
  - logical indicating if reflection is allowed

**Value**

A list with components:

- **R**
  - The estimated rotation matrix (may be an orthogonal matrix if reflection is allowed)

- **s**
  - The estimated scale matrix

- **Ahat**
  - The centred configuration A

- **Bhat**
  - The Procrustes registered configuration B

- **OSS**
  - The ordinary Procrustes sum of squares, which is $\|Ahat-Bhat\|^2$

- **rmsd**
  - rmsd = sqrt(OSS/(km))

**Author(s)**

Ian Dryden

**References**


**See Also**

procGPA, riemdist, tpsgrid
Examples

data(digit3.dat)
A<-digit3.dat[,1]
B<-digit3.dat[,2]
ans<-procOPA(A,B)
plotshapes(A,B,joinline=1:13)
plotshapes(ans$Ahat,ans$Bhat,joinline=1:13)

#Sooty Mangabey data
data(sooty.dat)
A<-sooty.dat[,1]  #juvenile
B<-sooty.dat[,2]  #adult
par(mfrow=c(1,3))
par(pty="s")
plot(A,xlim=c(-2000,3000),ylim=c(-2000,3000),xlab=" ",ylab=" ")
lines(A[c(1:12,1),])
points(B)
lines(B[c(1:12,1),],lty=2)
title("Juvenile (-------) Adult (- - - -)")

#match B onto A
out<-procOPA(A,B)
#rotation angle
print(atan2(out$R[1,2],out$R[1,1])*180/pi)
#scale
print(out$s)
plot(A,xlim=c(-2000,3000),ylim=c(-2000,3000),xlab=" ",ylab=" ")
lines(A[c(1:12,1),])
points(out$Bhat)
lines(out$Bhat[c(1:12,1),],lty=2)
title("Match adult onto juvenile")
#match A onto B
out<-procOPA(B,A)
#rotation angle
print(atan2(out$R[1,2],out$R[1,1])*180/pi)
#scale
print(out$s)
plot(B,xlim=c(-2000,3000),ylim=c(-2000,3000),xlab=" ",ylab=" ")
lines(B[c(1:12,1),],lty=2)
points(out$Bhat)
lines(out$Bhat[c(1:12,1),])
title("Match juvenile onto adult")

procWGPA  Weighted Procrustes analysis

Description

Weighted Procrustes analysis to register landmark configurations into optimal registration using translation, rotation and scaling. Registration without scaling is also an option. Also, obtains principal components, and some summary statistics.
**procWGPA**

**Usage**

```r
procWGPA(x, fixcovmatrix=FALSE, initial="Identity", maxiterations=10, scale=TRUE, reflect=FALSE, prior="Exponential", diagonal=TRUE, sampleweights="Equal")
```

**Arguments**

- **x**: Input k x m x n real array, where k is the number of points, m is the number of dimensions, and n is the sample size.
- **fixcovmatrix**: If FALSE then the landmark covariance matrix is estimated. If a fixed covariance matrix is desired then the value should be given here, e.g. `fixcovmatrix=diag(8)` for the identity matrix with 8 landmarks.
- **initial**: The initial value of the estimated covariance matrix. "Identity" - identity matrix, "Rawdata" - based on sample variance of the raw landmarks. Also, could be a k x k symmetric positive definite matrix.
- **maxiterations**: The maximum number of iterations for estimating the covariance matrix,
- **scale**: Logical quantity indicating if scaling is required,
- **reflect**: Logical quantity indicating if reflection invariance is required,
- **prior**: Indicates the type of prior. "Exponential" is exponential for the inverse eigenvalues. "Identity" is an inverse Wishart with the identity matrix as parameters.
- **diagonal**: Logical. Indicates if the diagonal of the landmark covariance matrix (only) should be used. Diagonal matrices can lead to some landmarks having very small variability, which may or may not be desirable.
- **sampleweights**: Gives the weights of the observations in the sample, rather than the landmarks. This is a fixed quantity. "Equal" indicates that all observations in the sample have equal weight. The weights do not need to sum to 1.

**Details**

The factored covariance model is assumed: $\Sigma_k \times I_m$ with $\Sigma_k$ being the covariance matrix of the landmarks, and the cov matrix at each landmark is the identity matrix.

**Value**

A list with components

- **k**: no of landmarks
- **m**: no of dimensions (m-D dimension configurations)
- **n**: sample size
- **mshape**: Weighted Procrustes mean shape.
- **tan**: This is the mk x n matrix of Procrustes residuals $X_i^P$ - Xbar.
- **rotated**: the k x m x n array of weighted Procrustes rotated data
- **pcar**: the columns are eigenvectors (PCs) of the sample covariance $S_v$ of tan
- **pcasd**: the square roots of eigenvalues of $S_v$ using tan (s.d.'s of PCs)
- **percent**: the percentage of variability explained by the PCs using tan.
qcet2.dat

T2 mouse vertebrae data - control group. 6 landmarks in 2 dimensions, 30 individuals

Usage
data(qcet2.dat)

Format
An array of dimension 6 x 2 x 30
Source


References

http://www.maths.nott.ac.uk/personal/ild/bookdata/qcet2.dat
Data from Paul O’Higgins (Hull-York Medical School) and David Johnson (Leeds)

Examples

data(qcet2.dat)
plotshapes(qcet2.dat)

qlet2.dat  Large T2 mouse vertabrae data

Description

T2 mouse vertebrae data - large group. 6 landmarks in 2 dimensions, 23 individuals

Usage

data(qlet2.dat)

Format

An array of dimension 6 x 2 x 23

Source


References

http://www.maths.nott.ac.uk/personal/ild/bookdata/qlet2.dat
Data from Paul O’Higgins (Hull-York Medical School) and David Johnson (Leeds)

Examples

data(qlet2.dat)
plotshapes(qlet2.dat)
**qset2.dat**  
*Small T2 mouse vertabre data*

**Description**
T2 mouse vertebrae data - small group. 6 landmarks in 2 dimensions, 23 individuals

**Usage**
data(qset2.dat)

**Format**
An array of dimension 6 x 2 x 23

**Source**

**References**
http://www.maths.nott.ac.uk/personal/ild/bookdata/qset2.dat  
Data from Paul O’Higgins (Hull-York Medical School) and David Johnson (Leeds)

**Examples**
data(qset2.dat)
plotshapes(qset2.dat)

---

**rats**  
*Rat skulls data*

**Description**
Rat skulls data, from X rays. 8 landmarks in 2 dimensions, 18 individuals observed at 7, 14, 21, 30, 40, 60, 90, 150 days.

**Usage**
data(rats)

**Format**
rats$x: An array of landmark configurations 144 x 2 x 2  
rats$no: Individual rat number (note rats 3, 13, 20 missing due to incomplete data)  
rats$time observed time in days
Source

Vilmann’s rat data set (Bookstein, 1991, Morphometric Tools for Landmark Data: Geometry and Biology, pp. 408-414)

References


Examples

data(rats)
plotshapes(rats$x, col=1:8)

resampletest(A, B, resamples = 200, replace = TRUE)
Arguments

A  The random sample for group 1: k x m x n1 array of data, where k is the number of landmarks and n1 is the sample size. (Alternatively a k x n1 complex matrix for 2D)

B  The random sample for group 3: k x m x n2 array of data, where k is the number of landmarks and n2 is the sample size. (Alternatively a k x n2 complex matrix for 2D)

resamples  Integer. The number of resampling iterations. If resamples = 0 then no resampling procedures are carried out, and the tabular p-values are given only.

replace  Logical. If replace = TRUE then for 2D data bootstrap resampling is carried out with replacement *within* each group. If replace = FALSE then permutation resampling is carried out (sampling without replacement in *pooled* samples).

Value

A list with components (or a subset of these)

lambda  $\lambda_{\text{min}}$ statistic

lambda.pvalue  p-value for $\lambda_{\text{min}}$ test based on resampling

lambda.table.pvalue  p-value for $\lambda_{\text{min}}$ test based on the asymptotic chi-squared distribution (large n1,n2)

H  The Hotelling $T^2$ statistic

H.pvalue  p-value for the Hotelling $T^2$ test based on resampling

H.table.pvalue  p-value for the Hotelling $T^2$ test based on the null F distribution, assuming normality and equal covariance matrices

J  The Hotelling $T^2$ statistic

J.pvalue  p-value for the Hotelling $T^2$ test based on resampling

J.table.pvalue  p-value for the Hotelling $T^2$ test based on the null F distribution, assuming normality and unequal covariance matrices

G  The Goodall F statistic

G.pvalue  p-value for the Goodall test based on resampling

G.table.pvalue  p-value for the Goodall test based on the null F distribution, assuming normality and equal isotropic covariance matrices

Author(s)

Ian Dryden

References

riemdist

Riemannian shape distance

Description

Calculates the Riemannian shape distance rho between two configurations

Usage

riemdist(x, y, reflect=FALSE)

Arguments

x  
k x m matrix (or complex k-vector for 2D data) where k = number of landmarks and m = no of dimensions

y  
k x m matrix (or complex k-vector for 2D data)

reflect  
Logical. If reflect = TRUE then reflection invariance is included.

Value

The Riemannian shape distance rho between the two configurations. Note 0 <= rho <= pi/2 if no reflection invariance. (for the Riemannian size-and-shape distance use ssriemdist)

See Also

testmeanshapes

Examples

#2D example : female and male Gorillas

data(gorf.dat)
data(gorm.dat)

#just select 3 landmarks and the first 10 observations in each group
select<-c(1,2,3)
A<-gorf.dat[select,,1:10]
B<-gorm.dat[select,,1:10]
resampletest(A,B,resamples=100)
rigidbody

Author(s)

Ian Dryden

References


See Also

procOPA, procGPA

Examples

data(gorf.dat)
data(gorm.dat)
gorf<-procGPA(gorf.dat)
gorm<-procGPA(gorm.dat)
rhoc<-riemdist(gorf$mshape,gorm$mshape)
cat("Riemannian distance between mean shapes is ",rho," \n")

rigidbody  Rigid body transformations

Description

Applies a rigid body transformations to a landmark configuration or array

Usage

rigidbody(X,transx=0,transy=0,transz=0,thetax=0,thetay=0,thetaz=0)

Arguments

X k x m matrix, or k x m x n array where k = number of landmarks and m = no of dimensions and n is no of specimens

transx negative shift in x-coordinates

transy negative shift in y-coordinates

transz negative shift in z-coordinates

thetax Rotation about x-axis in degrees

thetay Rotation about y-axis in degrees

thetaz Rotation about z-axis in degrees

Value

The transformed coordinates (X - trans) Rx Ry Rz
Author(s)
Ian Dryden

Examples
data(gorf.dat)
plotshapes ( rigidbody(gorf.dat , 0, 0, 0, 0, 0, -90 ) )

data(sand)
plotshapes(sand$x[,,sand$group=="sea"],sand$x[,,sand$group=="river"],joinline=c(1:50))

Description
50 points on 24 sea sand and 25 river sand grain profiles in 2D. The original data were kindly provided by Professor Dietrich Stoyan (Stoyan and Stoyan, 1994; Stoyan, 1997). The 50 points on each outline were extracted at approximately equal arc-lengths by the method described in Kent et al. (2000, section 8.1)

Usage
data(sand)

Format
A list with components:
sea$x : An array of dimension 50 x 2 x 49 containing the 50 point co-ordinates in 2D for each grain
sea$group : The types of the sand grains: "sea", 24 particles from the Baltic Sea
"river", 25 particles from the Caucasian River Selenchuk

References

Examples
data(sand)
plotshapes(sand$x[,sand$group=="sea"],sand$x[,sand$group=="river"],joinline=c(1:50))
**Description**

Bookstein’s schizophrenia data. 13 landmarks in 2 dimensions, 28 individuals. The first 14 individuals are controls. The last fourteen cases were diagnosed with schizophrenia. The landmarks were taken in the near midline from MR images of the brain: (1) splenium, posteriormost point on corpus callosum; (2) genu, anteriormost point on corpus callosum; (3) top of corpus callosum, uppermost point on arch of callosum (all three to an approximate registration on the diameter of the callosum); (4) top of head, a point relaxed from a standard landmark along the apparent margin of the dura; (5) tentorium of cerebellum at dura; (6) top of cerebellum; (7) tip of fourth ventricle; (8) bottom of cerebellum; (9) top of pons, anterior margin; (10) bottom of pons, anterior margin; (11) optic chiasm; (12) frontal pole, extension of a line from landmark 1 through landmark 2 until it intersects the dura; (13) superior colliculus.

**Usage**

`data(schizophrenia.dat)`

**Format**

`schizophrenia$x`: An array of dimension 13 x 2 x 28

`schizophrenia$group`: A factor of group labels ‘con’ for Controls and ‘scz’ for the schizophrenia patients.

**Source**


**References**

Data kindly provided by Fred Bookstein (University of Washington and University of Vienna)

**Examples**

```r
data(schizophrenia)
plotshapes(schizophrenia$x,symbol=as.integer(schizophrenia$group))```
**Description**

Bookstein’s schizophrenia data. 13 landmarks in 2 dimensions, 28 individuals. The first 14 individuals are controls. The last fourteen cases were diagnosed with schizophrenia. The landmarks were taken in the near midline from MR images of the brain: (1) splenium, posteriormost point on corpus callosum; (2) genu, anteriormost point on corpus callosum; (3) top of corpus callosum, uppermost point on arch of callosum (all three to an approximate registration on the diameter of the callosum); (4) top of head, a point relaxed from a standard landmark along the apparent margin of the dura; (5) tentorium of cerebellum at dura; (6) top of cerebellum; (7) tip of fourth ventricle; (8) bottom of cerebellum; (9) top of pons, anterior margin; (10) bottom of pons, anterior margin; (11) optic chiasm; (12) frontal pole, extension of a line from landmark 1 through landmark 2 until it intersects the dura; (13) superior colliculus.

**Usage**

```r
data(schizophrenia.dat)
```

**Format**

An array of dimension 13 x 2 x 28

**Source**


**References**

Data kindly provided by Fred Bookstein (University of Washington and University of Vienna)

**Examples**

```r
data(schizophrenia.dat)
k<-dim(schizophrenia.dat)[1]
n<-dim(schizophrenia.dat)[3]
plotshapes(schizophrenia.dat)
```
shapepca

**Principal components analysis for shape**

**Description**

Provides graphical summaries of principal components for shape.

**Usage**

```r
shapepca(proc, pcno = c(1, 2, 3), type = "r", mag = 1, joinline = c(1, 1),
project=c(1,2),scores3d=FALSE,color=2,axes3=FALSE,rglopen=TRUE,zslice=0)
```

**Arguments**

- **proc**
  - List given by the output from `procGPA()`
- **pcno**
  - A vector of the PCs to be plotted
- **type**
  - Options for the types of plot for the $m=2$ planar case: "r": rows along PCs evaluated at $c = -3, 0, 3$ sd’s along PC, "v": vectors drawn from mean to $+3$ sd’s along PC, "s": plots along $c = -3, -2, -1, 0, 1, 2, 3$ superimposed, "m": movie backward and forwards from $-3$ to $+3$ sd’s along PC, "g": TPS grid from mean to $+3$ sd’s along PC.
- **mag**
  - Magnification of the effect of the PC (scalar multiple of sd’s)
- **joinline**
  - A vector stating which landmarks are joined up by lines, e.g. `joinline=c(1:n,1)` will start at landmark 1, join to 2, ..., join to n, then re-join to landmark 1.
- **project**
  - The default orthogonal projections if in higher than 2 dimensions
- **scores3d**
  - Logical. If TRUE then a 3D scatterplot of the first 3 raw PC scores with labels in 'pcno' is given, instead of the default plot of the mean and PC vectors.
- **color**
  - Color of the spheres used in plotting. Default color = 2 (red). If a vector is given then the points are colored in that order.
- **axes3**
  - Logical. If TRUE then the axes are plotted in a 3D plot.
- **rglopen**
  - Logical. If TRUE open a new RGL window, otherwise plot in current window.
- **zslice**
  - For 3D case, type = "g": the z co-ordinate(s) for the grid slice(s)

**Details**

The mean and PCs are plotted.

**Value**

No value is returned

**Author(s)**

Ian Dryden
shapes.cva

Canonical variate analysis for shapes

Description

Carry out canonical variate analysis for shapes (in two or more groups)

Usage

`shapes.cva(X, groups, scale=TRUE, tangentcoords = "residual", ncv=2)`

Arguments

- **X**: Input k x m x n real array of the configurations, where k is the number of points, m is the number of dimensions, and n is the sample size.
- **groups**: The group labels
- **scale**: Logical, indicating if Procrustes scaling should be carried out
- **tangentcoords**: The type of Procrustes tangent coordinates to use (as for procGPA)
- **ncv**: Number of canonical variates to display

References


See Also

procGPA

Examples

```r
#2d example
data(gorf.dat)
data(gorm.dat)

gorf<-procGPA(gorf.dat)
gorm<-procGPA(gorm.dat)
shapepca(gorf,type="r",mag=3)
shapepca(gorf,type="v",mag=3)
shapepca(gorm,type="r",mag=3)
shapepca(gorm,type="v",mag=3)

#3D example
#data(macm.dat)
#out<-procGPA(macm.dat)
#movie
#shapepca(out,pcno=1)
```
Value
A plot if ncv=2 or 3 and the Canonical Variate Scores

Author(s)
Ian Dryden

References

See Also
procGPA

Examples

#2D example: female and male apes (cf. Dryden and Mardia, 2016)

data(pongof.dat)
data(pongom.dat)
data(panm.dat)
data(panf.dat)

apes <- groupstack( pongof.dat , pongom.dat , panm.dat, panf.dat )

shapes.cva( apes$x, apes$groups)

shapes3d

Description
Plot the landmark configurations from a 3D dataset

Usage
shapes3d(x, loop=0, type="p", color = 2, joinline=c(1:1), axes3=FALSE, rglopen=TRUE)

Arguments

x An array of size k x 3 x n, where k is the number of landmarks and n is the number of observations

loop gives the number of times an animated loop through the observations is displayed (in order 1 to n). loop > 0 is suitable when a time-series of shapes is available. loop = 0 gives a plot of all the observations on the same figure.
shells

**type**
Type of plot: "p" points, "dots" dots (quicker for large plots), "l" dots and lines though landmarks 1:k if 'joinline' not stated

**color**
Colour of points (default color = 2 (red)). If a vector is given then the points are coloured in that order.

**joinline**
Join the numbered landmarks by lines

**axes3**
Logical. If TRUE then plot the axes.

**rglopen**
Logical. If TRUE then open a new RGL window, if FALSE then plot in current window.

**Value**
None

**Author(s)**
Ian Dryden

**References**

**Examples**
```r
data(dna.dat)
shapes3d(dna.dat)
```

---

**shells**

Microfossil shell data

**Description**
Microfossil shell data. Triangles from 21 individuals. Lohmann (1983) published 21 mean outlines of the microfossil which were based on random samples of organisms taken at different latitudes in the South Indian Ocean.

**Usage**
```r
data(shells)
```

**Format**

- **shells$uv** Scaled shape coordinates (Bookstein shape co-ordinates with base (0,0) and (1,0).
- **shells$size** Centroid size
Source


References

The data have been extracted from Fig. 7 of Bookstein (1986).

Examples

data(shells)
plotshapes(shells$uv)

sooty
Sooty mangabey data

Description

Sooty mangabey data skull data. 12 landmarks in 2 dimensions, 2 individuals (juvenile and male adult) followed by three individuals, female adult, male adult. The first entries are rotated, translated versions of the 3rd and 7th figure.

Usage

data(sooty)

Format

An array of dimension 12 x 2 x 7

Source


References

Data from Paul O’Higgins (Hull-York Medical School)

Examples

data(sooty)
plotshapes(sooty, joinline=c(1:12, 1))
ssriemdist

Riemannian size-and-shape distance

Description
Calculates the Riemannian size-and-shape distance $d_S$ between two configurations

Usage
ssriemdist(x, y, reflect=FALSE)

Arguments
- **x**: k x m matrix (or complex k-vector for 2D data) where k = number of landmarks and m = no of dimensions
- **y**: k x m matrix (or complex k-vector for 2D data)
- **reflect**: Logical. If reflect = TRUE then reflection invariance is included.

Value
The Riemannian size-and-shape distance $d_S$ between the two configurations. (for the Riemannian shape distance use riemdist)

Author(s)
Ian Dryden

References

See Also
procOPA, procGPA, riemdist

Examples
data(gorf.dat)
data(gorm.dat)
gorf<-procGPA(gorf.dat, scale=FALSE)
gorm<-procGPA(gorm.dat, scale=FALSE)
ds<-ssriemdist(gorf$mshape, gorm$mshape)
cat("Riemannian size-and-shape distance between mean size-and-shapes is ", ds, " \n")
Description
Steroid data. Between 42 and 61 atoms for each of 31 steroid molecules.

Usage
data(steroids)

Format
steroids$x : An array of dimension 61 x 3 x 31 of 3D co-ordinates of the 31 steroids. If a molecules has less than 61 atoms then the remaining co-ordinates are all zero.
steroids$activity : Activity class (‘1’ = high, ‘2’ = intermediate, and ‘3’ = low binding affinities to the corticosteroid binding globulin (CBG) receptor)
steroids$radius : van der Waals radius (0 = missing value)
steroids$atom : atom type (0 = missing value)
steroids$charge : partial charge (0 = missing value)
steroids$names : steroid names

Source
This particular version of the steroids data set of (x, y, z) atom co-ordinates and partial charges was constructed by Jonathan Hirst and James Melville (School of Chemistry, University of Nottingham).
http://www2.ccc.uni-erlangen.de/services/steroids/

References

Examples
data(steroids)
shapes3d(steroids$x[,1])
**testmeanshapes**

*Tests for mean shape difference, including permutation and bootstrap tests*

**Description**

Carries out tests to examine differences in mean shape between two independent populations, for $m=2$ or $m=3$ dimensional data. Tests are carried out using tangent co-ordinates.

- **H**: Hotelling $T^2$ statistic (see Dryden and Mardia, 2016, equ.(9.4))
- **G**: Goodall’s F statistic (see Dryden and Mardia, 2016, equ.(9.9))
- **J**: James $T^2$ statistic (see Amaral et al., 2007)

P-values are given based on resampling (either a bootstrap test or a permutation test) as well as the usual table based p-values. Bootstrap tests involve sampling with replacement under H0 (as in Amaral et al., 2007).

Note when the sample sizes are low (compared to the number of landmarks) some minor regularization is carried out. In particular if $Sw$ is a singular within group covariance matrix, it is replaced by $Sw + 0.000001$ (Identity matrix) and a ‘*’ is printed in the output.

**Usage**

```r
testmeanshapes(A, B, resamples = 1000, replace = FALSE, scale = TRUE)
```

**Arguments**

- **A**: The random sample for group 1: $k \times m \times n_1$ array of data, where $k$ is the number of landmarks and $n_1$ is the sample size. (Alternatively a $k \times n_1$ complex matrix for 2D)
- **B**: The random sample for group 2: $k \times m \times n_2$ array of data, where $k$ is the number of landmarks and $n_2$ is the sample size. (Alternatively a $k \times n_2$ complex matrix for 2D)
- **resamples**: Integer. The number of resampling iterations. If resamples = 0 then no resampling procedures are carried out, and the tabular p-values are given only.
- **replace**: Logical. If replace = TRUE then bootstrap resampling is carried out with replacement *within* each group. If replace = FALSE then permutation resampling is carried out (sampling without replacement in *pooled* samples).
- **scale**: Logical. Whether or not to carry out Procrustes with scaling in the procedure.

**Value**

A list with components

- **H**: The Hotelling statistic (F statistic)
- **H.pvalue**: p-value for the Hotelling test based on resampling
- **H.table.pvalue**: p-value for the Hotelling test based on the null F distribution, assuming normality and equal covariance matrices
J The James $T^2$ statistic
J.pvalue p-value for the James $T^2$ test based on resampling
J.table.pvalue p-value for the James $T^2$ test based on the null F distribution, assuming
normality but unequal covariance matrices
G The Goodall $F$ statistic
G.pvalue p-value for the Goodall test based on resampling
G.table.pvalue p-value for the Goodall test based on the null F distribution, assuming normality
and equal isotropic covariance matrices

Author(s)

Ian Dryden

References

problems in directional statistics and shape analysis. Journal of the American Statistical Associa-
tion. 102, 695-707.

Dryden, I.L. and Mardia, K.V. (2016). Statistical Shape Analysis, with applications in R (Second


See Also

resampletest

Examples

#2D example : female and male Gorillas
data(gorf.dat)
data(gorm.dat)
A<-gorf.dat
B<-gorm.dat
testmeanshapes(A,B,resamples=100)


**Description**

Thin-plate spline transformation grids from one set of landmarks to another.

**Usage**

```r
tpsgrid(TT, YY, xbegin=-999, ybegin=-999, xwidth=-999, opt=1, ext=0.1, ngrid=22, cex=1, pch=20, col=2, zslice=0, mag=1, axes3=FALSE)
```

**Arguments**

- **TT**: First object (source): (k x m matrix)
- **YY**: Second object (target): (k x m matrix)
- **xbegin**: lowest x value for plot: if -999 then a value is determined
- **ybegin**: lowest y value for plot: if -999 then a value is determined
- **xwidth**: width of plot: if -999 then a value is determined
- **opt**: Option 1: (just deformed grid on YY is displayed), option 2: both grids are displayed
- **ext**: Amount of border on plot in 2D case.
- **ngrid**: Number of grid points: size is ngrid * (ngrid -1)
- **cex**: Point size
- **pch**: Point symbol
- **col**: Point colour
- **zslice**: For 3D case the scaled z co-ordinate(s) for the grid slice(s). The values are on a standardized scale as a proportion of height from the middle of the z-axis to the top and bottom. Values in the range -1 to 1 would be sensible.
- **mag**: Exaggerate effect (mag > 1). Standard effect has mag=1.
- **axes3**: Logical. If TRUE then the axes are plotted in a 3D plot.

**Details**

A square grid on the first configuration is deformed smoothly using a pair of thin-plate splines in 2D, or a triple of splines in 3D, to a curved grid on the second object. For 3D data the grid is placed at a constant z-value on the first figure, indicated by the value of zslice.

For 2D data the covariance function in the thin-plate spline is $\sigma(h) = |h|^2 \log |h|^2$ and in 3D it is given by $\sigma(h) = -|h|$.

**Value**

No returned value
transformations

Author(s)
Ian Dryden

References

See Also
procGPA

Examples

data(gorf.dat)
data(gorm.dat)

#TPS grid with shape change exaggerated (2x)
gorf<-procGPA(gorf.dat)
gorm<-procGPA(gorm.dat)
TT<-gorf$mshape
YY<-gorm$mshape
tpsgrid(TT,YY,mag=2)
title("TPS grid: Female mean (left) to Male mean (right)"")

transformations

Calculate similarity transformations

Description
Calculate similarity transformations between configurations in two arrays.

Usage
transformations(Xrotated,Xoriginal)

Arguments

Xrotated Input k x m x n real array of the Procrustes transformed configurations, where k is the number of points, m is the number of dimensions, and n is the sample size.

Xoriginal Input k x m x n real array of the Procrustes original configurations, where k is the number of points, m is the number of dimensions, and n is the sample size.
transformations

Value
A list with components
- **translation**: The translation parameters. These are the relative translations of the centroids of the individuals.
- **scale**: The scale parameters
- **rotation**: The rotation parameters. These are the rotations between the individuals after they have both been centred.

Author(s)
Ian Dryden

References

See Also
procGPA

Examples

```
#2D example : female and male Gorillas (cf. Dryden and Mardia, 2016)
data(gorf.dat)
Xorig <- gorf.dat
Xrotated <- procGPA(gorf.dat)$rotated
transformations(Xrotated,Xorig)
```
Index

* datasets
  apes, 3
  brains, 5
cortical, 7
digit3.dat, 8
dna.dat, 9
gels, 13
gorf.dat, 13
gorm.dat, 14
humanmove, 16
macaques, 17
macf.dat, 17
macm.dat, 18
mice, 19
panf.dat, 20
panm.dat, 20
pongof.dat, 27
pongom.dat, 28
qct2.dat, 36
qlet2.dat, 37
qset2.dat, 38
rats, 38
sand, 43
schizophrenia, 44
schizophrenia.dat, 45
shells, 49
sooty, 50
steroids, 52

* hplot
  plotshapes, 22
  shapepca, 46
tpsgrid, 55

* multivariate
  backfit, 3
  bookstein2d, 4
centroid.size, 6
distcov, 8
estcov, 10
frechet, 11
groups, 15
plot3darcs, 21
plotshapes, 22
pns, 23
pns4pc, 25
pns3d, 26
procdist, 29
procGPA, 30
procOPA, 32
procWGPA, 34
resampletest, 39
riemdists, 41
rigidbody, 42
shapepca, 46
shapes.cva, 47
shapes3d, 48
ssriemdist, 51
testmeanshapes, 53
tpsgrid, 55
transformations, 56

apes, 3
backfit, 3
bookstein2d, 4
brains, 5
centroid.size, 6
cortical, 7
digit3.dat, 8
distcov, 8
dna.dat, 9
estcov, 10
frechet, 11
gels, 13
gorf.dat, 13
gorm.dat, 14

58
INDEX

groupstack, 15
humanmove, 16
macaques, 17
macf.dat, 17
macm.dat, 18
mice, 19
panf.dat, 20
panm.dat, 20
plot3darcs, 21
plotshapes, 22
pns, 23
pns4pc, 25
pnss3d, 26
pongof.dat, 27
pongom.dat, 28
procdist, 29
procGPA, 30
procOPA, 32
procWGPA, 34
qcel2.dat, 36
qlet2.dat, 37
qset2.dat, 38
rats, 38
resampletest, 39
riemdist, 41
rigidbody, 42
sand, 43
schizophrenia, 44
schizophrenia.dat, 45
shapepca, 46
shapes.cva, 47
shapes3d, 48
shells, 49
sooty, 50
ssriemdist, 51
steroids, 52
testmeanshapes, 53
tpsgrid, 55
transformations, 56