Package ‘MultBiplotR’
September 29, 2016

Type Package
Title MULTivariate Analysis Using BIPlOTs
Version 0.3.3.1
Date 2015-01-13
Author Jose Luis Vicente-Villardon
Maintainer Jose Luis Vicente-Villardon <villardon@usal.es>
Description Several Multivariate Techniques from a biplot perspective. It is the translation into R of the previous package MultBiplot developed in Matlab.
License GPL (>=2)
Encoding latin1
Repository CRAN
Depends R (>= 3.2.0), scales, geometry, deldir, rgl, mirt, GPArotation
Imports MASS, lattice
LazyData yes
Archs i386, x64
NeedsCompilation no

R topics documented:

MultBiplotR-package ........................................... 4
AddCluster2Biplot ......................................... 5
AddContVars2Biplot ........................................ 7
AddSupVars2Biplot ......................................... 8
BinaryDistances ........................................... 9
BinaryProximities ......................................... 10
BootstrapDistance ......................................... 12
BootstrapScalar ........................................... 14
BootstrapSmacof ............................................ 16
CA .......................................................... 18
CanonicalBiplot ............................................ 19
CanonicalDistanceAnalysis ............................... 20
CanonicalStatsBiplot ...................................... 22
CategoricalDistances ....................................... 23
CategoricalProximities ................................... 24
CCA .......................................................... 25
R topics documented:

CheckBinaryMatrix .......................................................... 26
CheckBinaryVector ............................................................... 27
Chemical ................................................................. 28
Circle ................................................................. 29
ConcEllipse ............................................................... 29
ContinuousDistances .......................................................... 30
Convert2ThreeWay ............................................................ 32
ConvertFactors2Integers ..................................................... 33
CrissCross ................................................................. 33
CumSum ................................................................. 35
DataFrame2BinaryMatrix ..................................................... 36
DataFrame2Matrix4Regression ............................................... 37
DensityBiplot ............................................................... 37
dhats ................................................................. 38
diagonal .............................................................. 39
dimensionLabels ............................................................. 40
dlines ................................................................. 40
Doctors ................................................................. 41
EuclideanDistance ............................................................. 42
ExpandTable ................................................................. 43
ExternalBinaryLogisticBiplot ............................................... 43
ExtractTable ............................................................... 45
FA.Biplot ................................................................. 46
Factor2Binary ............................................................... 48
Fraction ................................................................. 49
GeneralizedProcrustes ......................................................... 50
getBiplotScales ............................................................. 51
getCCAScales ............................................................... 52
ginv ................................................................. 53
GowerProximities ............................................................ 54
GowerSimilarities ............................................................ 55
Hermquad ................................................................. 56
HJ.Biplot ................................................................. 56
InBox ................................................................. 58
InitialTransform ............................................................. 59
Integer2Binary .............................................................. 60
LogFrequencyBiplot .......................................................... 61
logit ................................................................. 63
Matrix2Proximities ............................................................ 64
matrixsqrt .............................................................. 65
matrixsqrtinv ............................................................. 66
MDS ................................................................. 67
MGC ................................................................. 69
MonotoneRegression ........................................................... 70
moth ................................................................. 71
Multiquad ................................................................. 72
MultiTableStatistics .......................................................... 72
MultiTableTransform ........................................................ 73
NiceNumber ................................................................. 74
NominalDistances ............................................................. 75
Numeric2Binary .............................................................. 76
ones ................................................................. 77
R topics documented:

OrdinalLogisticFit ........................................... 78
OrdLogBipEM .................................................... 79
OrdVarBiplot ................................................... 81
OrdVarCoordinates ............................................ 82
OrthogonalizeScores ......................................... 83
PCA.Biplot ...................................................... 84
plot.CA.sol ...................................................... 86
plot.Canonical.Biplot ......................................... 87
plot.CCA.sol ................................................... 90
plot.ContinuousBiplot ........................................ 92
plot.ellipse .................................................... 95
plot.fraction .................................................. 98
plot.MGC ....................................................... 99
plot.Ordinal.Logistic.Biplot ................................ 100
plot.PCoABootstrap .......................................... 102
plot.Principal.Coordinates ................................ 103
plot.Procrustes .............................................. 105
plot.Statis ..................................................... 106
plot.Unfolding ............................................... 107
plot3d.ContinuousBiplot ..................................... 108
plot3dCanonicalBiplot ....................................... 110
PlotBiplotClusters ........................................... 112
PlotOrdinalResponses ........................................ 113
PLSRfit ........................................................ 114
PoliticalFigures .............................................. 115
PrettyTicks .................................................... 116
PrincipalCoordinates ........................................ 117
print.MGC ..................................................... 118
Protein ........................................................ 119
RAPD ........................................................... 120
RemoveRowsWithNaNs ........................................ 121
riano ............................................................. 121
RidgeBinaryLogistic .......................................... 122
RidgeBinaryLogisticFit ...................................... 125
RidgeMultinomialLogisticFit ................................. 126
RidgeMultinomialLogisticRegression ....................... 128
RidgeOrdinalLogistic ......................................... 129
scores.CCA.sol ................................................ 131
SeparateVarTypes ............................................. 132
SimpleProcrustes ............................................. 133
SMACOF ........................................................ 134
smoking ......................................................... 136
spiders ........................................................ 137
SpidersEnv ...................................................... 138
SpidersSp ....................................................... 139
StatisBiplot .................................................... 140
summary.Canonical.Biplot .................................. 141
summary.CCA.sol ............................................. 142
summary.ContinuousBiplot .................................. 142
summary.MGC .................................................. 143
summary.Principal.Coordinates ............................. 144
MultBiplotR-package

Description

Classical PCA biplot with additional features as non-standard data transformations, scales for the variables, together with many graphical aids as sizes or colors of the points according to their qualities of representation or predictiveness. The package includes also Alternating Least Squares (ALS) or Criss-Cross procedures for the calculation of the reduced rank approximation that can deal with missing data, differential weights for each element of the data matrix or even robust versions of the procedure.

This is part of a bigger project called MultibiplotR that contains many other biplot techniques and is a translation to R of the package MULTIBILOT programmed in MATLAB. A GUI for the package is also in preparation.

Details

Package: MultBiplot
Type: Package
Version: 0.1.00
Date: 2015-01-14
License: GPL(>=2)

Author(s)

Jose Luis Vicente Villardon Maintainer: Jose Luis Vicente Villardon <villardon@usal.es>

References

AddCluster2Biplot

Add clusters to a biplot object

Description

The function add clusters to a biplot object to be represented on the biplot. The clusters can be defined by a nominal variable provided by the user, obtained from the hclust function of the base package or from the kmeans function.

Usage

AddCluster2Biplot(Bip, NGroups=3, ClusterType="hi", Groups=NULL, Original=FALSE, ...)

Examples

data(iris)
bip=PCA.Biplot(iris[,1:4])
plot(bip)
**Arguments**

- **Bip**
  A Biplot object obtained from any biplot procedure. It has to be a list containing a field called `Bip$RowCoordinates` in order to calculate the clusters when necessary.

- **NGroups**
  Number of groups or clusters. Only necessary when hierarchical or k-means procedures are used.

- **ClusterType**
  The type of cluster to add. There are three possibilities "us" (User Defined), "hi" (hierarchical clusters), "km" (kmeans clustering) or "gm" (gaussian mixture).

- **Groups**
  A factor defining the groups provided by the user.

- **Original**
  Should the clusters be calculated using the original data rather than the reduced dimensions?

- **...**
  Any other parameter for the `hclust` and `kmeans` procedures.

**Details**

One of the main shortcomings of cluster analysis is that it is not easy to search for the variables associated to the obtained classification; representing the clusters on the biplot can help to perform that interpretation. If you consider the technique for dimension reduction as a way to separate the signal from the noise, clusters should be constructed using the dimensions retained in the biplot, otherwise the complete original data matrix can be used. The colors used by each cluster should match the color used in the Dendrogram. User defined clusters can also be plotted, for example, to investigate the relation of the biplot solution to an external nominal variable.

**Value**

The function returns the biplot object with the information about the clusters added in new fields:

- **ClusterType**
  The method of clustering as defined in the argument `ClusterType`.

- **Clusters**
  A factor containing the solution or the user defined clusters.

- **ClusterNames**
  The names of the clusters.

- **ClusterColors**
  The colors of the clusters.

- **Dendrogram**
  The Dendrogram if we have used hierarchical clustering.

- **ClusterObject**
  The object obtained from `hclust`, `kmeans` or `MGC`.

**Author(s)**

Jose Luis Vicente Villardon

**References**


See Also

For clusters not provided by the user the function uses the standard procedures in hclust and kmeans.

Examples

data(Protein)
bip=PCA.Biplot(Protein[,3:11])
plot(bip)
# Add user defined clusters containing the region (North, South, Center)
bip=AddCluster2Biplot(bip, ClusterType="us", Groups=Protein$Region)
plot(bip, mode="a", margin=0.1, PlotClus=TRUE)
## Not run:
# Hierarchical clustering on the biplot coordinates using the Ward method
bip=AddCluster2Biplot(bip, ClusterType="hi", method="ward.D")
op <- par(mfrow=c(1,2))
plot(bip, mode="s", margin=0.1, PlotClus=TRUE)
plot(bip$endrogram)
par(op)
# K-means cluster on the biplot coordinates using the Ward method
bip=AddCluster2Biplot(bip, ClusterType="hi", method="ward.D")
op <- par(mfrow=c(1,2))
plot(bip, mode="s", margin=0.1, PlotClus=TRUE)
plot(bip$endrogram)
par(op)
## End(Not run)
Value
A biplot object with the coordinates for the supplementary variables added.

Author(s)
Jose Luis Vicente Villardon

See Also
addsupvarsRbiplot

Examples

## Should be DIRECTLY executable !! ----

Description
Adds supplementary variables to a biplot object constructed with any of the biplot methods of the package. The new variables are fitted using the coordinates for the rows. Each variable is fitted using the adequate procedure for its type.

Usage
AddSupVars2Biplot(bip, X)

Arguments
bip  The biplot object
X   A data frame with the supplementary variables.

Details
Binary, nominal or ordinal variables are fitted using logistic biplots. Continuous variables are fitted with linear regression.

Value
A biplot object with the coordinates for the supplementary variables added.

Author(s)
Jose Luis Vicente Villardon

See Also
AddContVars2Biplot
Binary Distances

Examples

```#
 Should be DIRECTLY executable !! ----
```

---

**Description**

Calculates distances among rows of a binary data matrix or among the rows of two binary matrices. The end user will use BinaryProximities rather than this function. Input must be a matrix with 0 or 1 values.

**Usage**

```r
BinaryDistances(x, y = NULL, coefficient= "Simple_Matching", transformation="sqrt(1-S)")
```

**Arguments**

- `x`: Main binary data matrix. Distances among rows are calculated if `y=NULL`.
- `y`: Second binary data matrix. If not `NULL` the distances among the rows of `x` and `y` are calculated.
- `coefficient`: Similarity coefficient. Use the name (see details).
- `transformation`: Transformation of the similarities. Use the name (see details).

**Details**

The following coefficients are calculated:

1. Kulezynski = $a/(b + c)$
2. Russell_and_Rao = $a/(a + b + c+d)$
3. Jaccard = $a/(a + b + c)$
4. Simple_Matching = $(a + d)/(a + b + c + d)$
5. Anderberg = $a/(a + 2 * (b + c))$
6. Rogers_and_Tanimoto = $(a + d)/(a + 2 * (b + c) + d)$
7. Sorensen_Dice_and_Czekanowski = $a/(a + 0.5 * (b + c))$
8. Sneath_and_Sokal = $(a + d)/(a + 0.5 * (b + c) + d)$
9. Hamman = $(a - (b + c) + d)/(a + b + c + d)$
10. Kulezynski = $0.5 * ((a/(a+b)) + (a/(a+c)))$
11. Anderberg2 = $0.25 * (a/(a+b) + a/(a+c) + d/(c+d) + d/(b+d))$
12. Ochiai = $a/sqrt((a + b) * (a + c))$
13. S13 = $(a * d)/sqrt((a + b) * (a + c) * (d + b) * (d + c))$
14. Pearson_phi = $(a * d - b * c)/sqrt((a + b) * (a + c) * (d + b) * (d + c))$
15. Yule = $(a * d - b * c)/(a * d + b * c)$

The following transformations of the similarity are calculated:

1. 'Identity' dis=sim
2. `1-S` dis = 1 - sim
3. `sqrt(1-S)` dis = sqrt(1 - sim)
4. `log(s)` dis = 1 - log(sim)
5. `1/S-1` dis = 1/sim -1
6. `sqrt(2(1-S))` dis = sqrt(2 * (1 - sim))
7. `1-(S+1)/2` dis = 1 - (sim + 1)/2
8. `1-abs(S)` dis = 1 - abs(sim)
9. `1/(S+1)` dis = 1/(sim) + 1

Value

An object of class `proximities`. This has components:

- `comp1` Description of `comp1`

Author(s)

Jose Luis Vicente-Villardon

References


See Also

`principalcoordinates`

Examples

data(spiders)

---

**Description**

Calculation of proximities among rows or columns of a binary data matrix or a data frame that will be converted into a binary data matrix.

**Usage**

`BinaryProximities(x, y = NULL, coefficient = "Jaccard", transformation = NULL, transpose = FALSE, ...)"`
**BinaryProximities**

**Arguments**

- **x**: A data frame or a binary data matrix. Proximities among the rows of x will be calculated.
- **y**: Supplementary data. The proximities among the rows of x and the rows of y will be also calculated.
- **coefficient**: Similarity coefficient. Use the number or the name (see details).
- **transformation**: Transformation of the similarities. Use the number or the name (see details).
- **transpose**: Logical. If TRUE, proximities among columns are calculated.
- ... Used to provide additional parameters for the conversion of the dataframe into a binary matrix.

**Details**

A binary data matrix is a matrix with values 0 or 1 coding the absence or presence of several binary characters. When a data frame is provided, every variable in the data frame is converted to a binary variable using the function `dataframe2BinaryMatrix`. Factors with two levels are converted directly to binary variables, factors with more than two levels are converted to a matrix with as many columns as levels and numerical variables are converted to binary variables using a cut point that can be the median, the mean or a value provided by the user.

The following coefficients are calculated:

1. Kulezynski = \( \frac{a}{b + c} \)
2. Russell and Rao = \( \frac{a}{a + b + c + d} \)
3. Jaccard = \( \frac{a}{a + b + c} \)
4. Simple Matching = \( \frac{a + d}{a + b + c + d} \)
5. Anderberg = \( \frac{a}{a + 2(b + c)} \)
6. Rogers and Tanimoto = \( \frac{a + d}{a + 2(b + c) + d} \)
7. Sorensen Dice and Czekanowski = \( \frac{a}{a + 0.5(b + c)} \)
8. Sneath and Sokal = \( \frac{a + d}{a + 0.5(b + c) + d} \)
9. Hamman = \( \frac{a - (b + c) + d}{a + b + c + d} \)
10. Kulezynski = \( 0.5 \times (a/(a + b) + a/(a + c)) \)
11. Anderberg2 = \( 0.25 \times (a/(a + b) + a/(a + c) + d/(c + d) + d/(b + d)) \)
12. Ochiai = \( a/sqrt((a + b) \times (a + c)) \)
13. S13 = \( (a \times d)/sqrt((a + b) \times (a + c) \times (d + b) \times (d + c)) \)
14. Pearson phi = \( (a \times d - b \times c)/sqrt((a + b) \times (a + c) \times (d + b) \times (d + c)) \)
15. Yule = \( (a \times d - b \times c)/(a \times d + b \times c) \)

The following transformations of the similarity3 are calculated:

1. 'Identity' \( dis=sim \)
2. '1-S' \( dis=1-sim \)
3. 'sqrt(1-S)' \( dis=sqrt(1 - sim) \)
4. '-log(s)' \( dis=-1*log(sim) \)
5. '1/S-1' \( dis=1/sim -1 \)
6. 'sqrt(2(1-S))' \( dis== sqrt(2*(1 - sim)) \)
7. '1-(S+1)/2' \( dis=1-(sim+1)/2 \)
8. \( '1-\text{abs}(S)' \text{ dis}=1-\text{abs}(\text{sim}) \)
9. \( '1/(S+1)' \text{ dis}=1/(\text{sim})+1 \)

Note that, after transformation the similarities are converted to distances except for "Identity". Not all the transformations are suitable for all the coefficients. Use them at your own risk. The default values are admissible combinations.

Value

An object of class `proximities`. This has components:

- **Type**
  - Binary, Continuous or Mixed. Binary in this case.
- **Coefficient**
  - Coefficient used to calculate the proximities
- **Transformation**
  - Transformation used to calculate the proximities
- **Data**
  - Data used to calculate the proximities
- **SupData**
  - Supplementary Data, if any
- **Proximities**
  - Proximities among rows of \( x \). May be similarities or dissimilarities depending on the transformation
- **SupProximities**
  - Proximities among rows of \( x \) and \( y \).

Author(s)

Jose Luis Vicente-Villardon

References


See Also

`BinaryDistances`, `Dataframe2BinaryMatrix`

Examples

```r
data(spiders)
D=BinaryProximities(spiders, coefficient="Jaccard", transformation="sqrt(1-S)")
D2=BinaryProximities(spiders, coefficient=3, transformation=3)
```

Description

Obtains bootstrap replicates of a distance matrix using random samples or permutations of the residual matrix from a Principal Coordinates (Components) Analysis. The object is to estimate the sampling variability of absorbed variances, coordinates and qualities of representation in a PCoA.
**BootstrapDistance**

**Usage**

BootstrapDistance(D, W=diag(nrow(D)), nB=200, dimsol=2,  
ProcrustesRot=TRUE, method=c("Sampling", "Permutation"))

**Arguments**

- **D** A distance matrix
- **W** A diagonal matrix containing weights for the rows of D
- **nB** Number of Bootstrap replications
- **dimsol** Dimension of the solution
- **ProcrustesRot** Should each replication be rotated to match the initial solution?
- **method** The replications are obtained "Sampling" or "Permutating" the residuals.

**Details**

The function calculates bootstrap confidence intervals for the inertia, coordinates and qualities of representation of a Principal Coordinates Analysis using a distance matrix as a basis. The function uses random sampling or permutations of the residuals to obtain the bootstrap replications. The procedure preserves the length of the points in the multidimensional space perturbing only the angles among the vectors. It is done so to preserve the property of positiveness of the diagonal elements of the scalar product matrices. The procedure may result into a scalar product that does not have an euclidean configuration and then has some negative eigenvalues; to avoid this problem the negative eigenvalues are removed to approximate the perturbated matrix by the closest with the required properties.

It is well known that the eigenvectors of a matrix are unique except for reflections, that is, if we change the sign of each component of the eigenvector we have the same solution. If that happens, an unwanted increase in the variability due to this artifact may invalidate the results. To avoid this we can calculate the scalar product of each eigenvector of the initial matrix with the corresponding eigenvector of the bootstrap replicate and change the signs of the later if the result is negative.

Another artifact of the procedure may arise when the dimension of the solution is higher than 1 because the eigenvectors of a replicate may generate the same subspace although are not in the same directions, i. e., the subspace is referred to a different system. That also may produce an unwanted increase of the variability that invalidates the results. To avoid this, every replicate may be rotated to match as much as possible the subspace generated by the eigenvectors of the initial matrix. This is done by Procrustes Analysis, taking the rotated matrix as solution. The solution to this problem is also a solution to the reflection, then only this problem is considered.

**Value**

Returns an object of class "PCoABootstrap" with the information for each bootstrap replication.

- **Eigenvalues** A matrix with dimensions in rows and replicates in columns containing the eigenvalues of each replicate in columns
- **Inertias** A matrix with dimensions in rows and replicates in columns containing the inertias of each replicate in columns
- **Coordinates** A list with a component for each object. A component contains the coordinates of an object for each replicate (in columns)
- **Values-Table** A list with a component for each object. A component contains the qualities of an object for each replicate (in columns)
- **NReplicates** Number of bootstrap replicates
**BootstrapScalar**

**Author(s)**

Jose L. Vicente-Villardon, Jhonny R. Demey

**References**


**See Also**

BootstrapScalar, ~~~

**Examples**

data(spiders)
D=BinaryProximities(spiders, coefficient="Jaccard", transformation="sqrt(1-S)")
DB=BootstrapDistance(D$Proximities)

---

**BootstrapScalar**

Bootstrap on the scalar product matrices used for Principal Coordinates Analysis (PCoA)

**Description**

Obtains bootstrap replicates of a scalar products matrix using random samples or permutations of the residual matrix from a Principal Coordinates (Components) Analysis. The object is to estimate the sampling variability of absorbed variances, coordinates and qualities of representation in a PCoA.

**Usage**

```r
BootstrapScalar(B, W=diag(nrow(B)), nB=200, dimsol=2, ProcrustesRot=TRUE, method=c("Sampling", "Permutation"))
```

**Arguments**

- `B`: A scalar product matrix
- `W`: A diagonal matrix containing waiths for the rows of `D`
- `nB`: Number of Bootstrap replications
- `dimsol`: Dimension of the solution
- `ProcrustesRot`: Should each replication be rotated to match the initial solution?
- `method`: The replications are obtained "Sampling" or "Permutating" the residuals.
Details

The function calculates bootstrap confidence intervals for the inertia, coordinates and qualities of representation of a Principal Coordinates Analysis using a distance matrix as a basis. The function uses random sampling or permutations of the residuals to obtain the bootstrap replications. The procedure preserves the length of the points in the multidimensional space perturbing only the angles among the vectors. It is done so to preserve the property of positiveness of the diagonal elements of the scalar product matrices. The procedure may result into a scalar product that does not have an euclidean configuration and then has some negative eigenvalues; to avoid this problem the negative eigenvalues are removed to approximate the perturbated matrix by the closest with the required properties.

It is well known that the eigenvectors of a matrix are unique except for reflections, that is, if we change the sign of each component of the eigenvector we have the same solution. If that happens, an unwanted increase in the variability due to this artifact may invalidate the results. To avoid this we can calculate the scalar product of each eigenvector of the initial matrix with the corresponding eigenvector of the bootstrap replicate and change the signs of the later if the result is negative.

Another artifact of the procedure may arise when the dimension of the solution is higher than 1 because the eigenvectors of a replicate may generate the same subspace although are not in the same directions, i. e., the subspace is referred to a different system. That also may produce an unwanted increase of the variability that invalidates the results. To avoid this, every replicate may be rotated to match as much as possible the subspace generated by the eigenvectors of the initial matrix. This is done by Procrustes Analysis, taking the rotated matrix as solution. The solution to this problem is also a solution to the reflection, then only this problem is considered.

Value

Returns an object of class "PCoABootstrap" with the information for each bootstrap replication.

Eigenvalues A matrix with dimensions in rows and replicates in columns containing the eigenvalues of each replicate in columns

Inertias A matrix with dimensions in rows and replicates in columns containing the inertias of each replicate in columns

Coordinates A list with a component for each object. A component contains the coordinates of an object for each replicate (in columns)

Values-Table A list with a component for each object. A component contains the qualities of an object for each replicate (in columns)

NReplicates Number of bootstrap replicates

Author(s)

Jose L. Vicente-Villardon, Jhonny R. Demey

References


See Also

BootstrapScalar.

Examples

```r
data(spiders)
D=BinaryProximities(spiders, coefficient="Jaccard", transformation="sqrt(1-S)"
)n=nrow(D$Proximities)
B=-0.5*(diag(n)-matrix(1,n,n)/n)
DB=BootstrapScalar(B)
```
Details

The function calculates bootstrap confidence intervals for coordinates and different stress measures using a distance matrix as a basis. The function uses random sampling or permutations of the residuals to obtain the bootstrap replications. The procedure preserves the length of the points in the multidimensional space perturbing only the angles among the vectors. It is done so to preserve the property of positiveness of the diagonal elements of the scalar product matrices. The procedure may result into a scalar product that does not have an euclidean configuration and then has some negative eigenvalues; to avoid this problem the negative eigenvalues are removed to approximate the perturbated matrix by the closest with the required properties.

It is well known that the eigenvectors of a matrix are unique except for reflections, that is, if we change the sign of each component of the eigenvector we have the same solution. If that happens, an unwanted increase in the variability due to this artifact may invalidate the results. To avoid this we can calculate the scalar product of each eigenvector of the initial matrix with the corresponding eigenvector of the bootstrap replicate and change the signs of the later if the result is negative.

Another artifact of the procedure may arise when the dimension of the solution is higher than 1 because the eigenvectors of a replicate may generate the same subspace although are not in the same directions, i.e., the subspace is referred to a different system. That also may produce an unwanted increase of the variability that invalidates the results. To avoid this, every replicate may be rotated to match as much as possible the subspace generated by the eigenvectors of the initial matrix. This is done by Procrustes Analysis, taking the rotated matrix as solution. The solution to this problem is also a solution to the reflection, then only this problem is considered.

Value

Returns an object of class "PCoABootstrap" with the information for each bootstrap replication.

Info

Information about the procedure

InitialDistance

Initial distance

RawStress

A vector containing the raw stress for all the bootstrap replicates

stress1

A vector containing the value of the stress1 formula for all the bootstrap replicates

stress2

A vector containing the value of the stress2 formula for all the bootstrap replicates

sstress1

A vector containing the value of the sstress1 formula for all the bootstrap replicates

sstress2

A vector containing the value of the sstress2 formula for all the bootstrap replicates

Coordinates

A list with a component for each object. A component contains the coordinates of an object for all the bootstrap replicates (in columns)

NReplicates

Number of bootstrap replicates

Author(s)

Jose L. Vicente-Villardon, Jhonny R. Demey
References


See Also

BootstrapScalar

Examples

data(spiders)
D=BinaryProximities(spiders, coefficient="Jaccard", transformation="sqrt(1-S)")
DB=BootstrapDistance(D$Proximities)

Correspondence Analysis

Description

Correspondence Analysis for a frequency or abundance data matrix.

Usage

CA(x, dim = 2, alpha = 1)

Arguments

x The frequency or abundance data matrix.
dim Dimension of the final solution
alpha Alpha to determine the kind of biplot to use.

Details

Calculates Correspondence Analysis for a two-way frequency or abundance table

Value

Correspondence analysis solution

Author(s)

Jose Luis Vicente Villardon
**CanonicalBiplot**

**References**


**Examples**

```r
data(riano)
Sp=riano[,3:15]
Env=riano[,16:25]
cabip=CA(Sp)
```

**CanonicalBiplot**  
*Biplot representation of a Canonical Variate Analysis or a Manova (Canonical-Biplot or MANOVA-Biplot)*

**Description**

Calculates a canonical biplot with confidence regions for the means.

**Usage**

```r
CanonicalBiplot(X, group, SUP = NULL, InitialTransform = 5)
```

**Arguments**

- **X**: A data matrix
- **group**: A factor containing the groups
- **SUP**: Supplementary observations to project on the biplot
- **InitialTransform**: Initial transformation of the data matrix

**Details**

The Biplot method (Gabriel, 1971; Galindo, 1986; Gower and Hand, 1996) is becoming one of the most popular techniques for analysing multivariate data. Biplot methods are techniques for simultaneous representation of the $n$ rows and $n$ columns of a data matrix $X$, in reduced dimensions, where the rows represent individuals, objects or samples and the columns the variables measured on them. Classical Biplot methods are a graphical representation of a Principal Components Analysis (PCA) that it is used to obtain linear combinations that successively maximize the total variability. PCA is not considered an appropriate approach where there is known a priori group structure in the data. The most general methodology for discrimination among groups, using multiple observed variables, is Canonical Variate Analysis (CVA). CVA allows us to derive linear combinations that successively maximize the ratio of "between-groups" to "pooled within-group" sample variance. Several authors propose a Biplot representation for CVA called Canonical Biplot (CB) (Vicente-Villardon, 1992 and Gower & Hand, 1996) when it is oriented to the discrimination between groups or MANOVA-Biplot Gabriel (1972, 1995) when the aim is to study the variables responsible for the discrimination. The main advantage of the Biplot version of the technique is that it is possible not only to establish the differences between groups but also to characterise the variables responsible for them. The methodology is not yet widely used mainly because it is still not available in the major statistical packages. Amaro, Vicente-Villardon & Galindo (2004) extend the methodology for two-way designs and propose confidence circles based on univariate and multivariate tests to perform post-hoc analysis of each variable.
CanonicalDistanceAnalysis

Value
An object of class "Canonical.Biplot"

Author(s)
Jose Luis Vicente Villardon

References

Examples

data(wine)
x=wine[,4:21]
canbip=CanonicalBiplot(x, group=wine$Group)
plot(canbip, mode="s")

CanonicalDistanceAnalysis

MANOVA and Canonical Analysis of Distances

Description
Performs a MANOVA and a Canonical Analysis based on of Distance Matrices (usually for continuous data)

Usage
CanonicalDistanceAnalysis(Prox, group, dimens = 3, Nsamples = 1000, PCoA = "Standard", ProjectInd =
**Arguments**

Prox        A object containing proximities
group    A factor with the group structure of the rows
dimens   The dimension of the solution
Nsamples Number of samples for the permutation test. Number of permutations.
PCoA      Type of Principal Coordinates for the Canonical Analysis calculated from the Principal coordinates of the Mean Matrix. "Standard" : Standard Principal Coordinates Analysis, "Weighted": Weighted Principal Coordinates Analysis, "WPCA")
ProjectInd Should the individual points be Projected onto the representation For the moment only available for Continuous Data.

**Details**

Performs a MANOVA and a Canonical Analysis based on of Distance Matrices (usually for continuous data). The MANOVA statistics is calculated from a decomposition of the distance matrix based on a factor of a external classification. The significance of the test is calculated using a permutation test. The approach depends only on the distances and then is useful with any kind of data.

The Canonical Representation is calculated from a Principal Coordinates Analysis od the distance matrix among the means. Thus, it is only possible for continuous data. The PCoA representation can be "Standard" using the means directly, "Weighted" weighting each group with its sample size or using weighted Principal Components Analysis of the matrix of means.

A measure of the quality of representation of the groups. When possible, the measure is also provided for the individual points.

Soon, a biplot representation will also be provided.

**Value**

An object of class 'CanonicalDistanceAnalysis" with:

Distances The Matrix of Distances from which the Analysis has been made
Groups A factor containing the group structure of the individuals
TSS            Total sum of squares
BSS            Between groups sum of squares
WSS            Within groups sum of squares
Fexp        Experimental pseudo F-value
pvalue p value based on the permutation test
Nsamples p value based on the permutation test
ExplainedVariance Variances explained by the PCoA
MeanCoordinates Coordinates of the groups for the graphical representation
Qualities Qualities of the representation of the groups
CumulativeQualities Cumulative qualities of the representation of the groups
RowCoordinates Coordinates of the individuals for the graphical representation
Note
The MANOVA and the representation of the means can be applied to any Distance although the projection of the individuals can be made only for continuous data.

Author(s)
Jose Luis Vicente Villardon

References

See Also
selectvarsAnova

Examples
```r
data(iris)
group=iris[,5]
x=as.matrix(iris[,1:4])
D=ContinuousProximities(x, coef = 1)
CDA=CanonicalDistanceAnalysis(D, group, dimens=2)
summary(CDA)
```

---

**CanonicalStatisBiplot**

*CANONICAL STATIS-ACT for multiple tables with common rows and its associated Biplot*

**Description**

The procedure performs STATIS-ACT methodology for multiple tables with common rows and its associated biplot

**Usage**

```r
StatisBiplot(X, Groups, InitTransform = "Standardize columns", dimens = 2,
             SameVar = FALSE)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>A list containing multiple tables with common rows</td>
</tr>
<tr>
<td>Groups</td>
<td>A factor containing the groups</td>
</tr>
<tr>
<td>InitTransform</td>
<td>Initial transformation of the data matrices</td>
</tr>
<tr>
<td>dimens</td>
<td>Dimension of the final solution</td>
</tr>
<tr>
<td>SameVar</td>
<td>Are the variables the same for all occasions?</td>
</tr>
</tbody>
</table>
Details

The procedure performs STATIS-ACT methodology for multiple tables with common rows and its associated biplot. When the variables are the same for all occasions trajectories for the variables can also be plotted.

Value

An object of class StatisBiplot

Author(s)

Jose Luis Vicente Villardon

References


Examples

data(Chemical)
x = Chemical[37:144,5:9]
weeks = as.factor(as.numeric(Chemical$WEEKS[37:144]))
levels(weeks) = c("W2", "W3", "W4")
X = Convert2ThreeWay(x, weeks, columns=FALSE)
groups = Chemical$Treatment[1:36]
canstbip = CanonicalStatisBiplot(X, groups, SameVar = TRUE)
plot(canstbip, mode="s", PlotVars=TRUE, ShowBox=TRUE)

CategoricalDistances

Distances among individuals using nominal variables.

Description

Distances among individuals using nominal variables.

Usage

CategoricalDistances(x, y = NULL, coefficient = "GOW", transformation = "sqrt(1-S)")
CategoricalProximities

Arguments

\(x\)  \quad \text{Matrix of Categorical Data}
\(y\)  \quad \text{A second matrix of categorical data with the same variables as } x
coefficient  \quad \text{Similarity coefficient to use (see details)}
transformation  \quad \text{Transformation of the similarity into a distance}

Details

The function calculates similarities and dissimilarities among a set of objects characterized by a set of nominal variables. The function uses similarities and converts into dissimilarities using a variety of transformations controlled by the user.

Value

A matrix with distances among the rows of \(x\) and \(y\). If \(y\) is NULL the interdistances among the rows of \(x\) are calculated.

Author(s)

Jose Luis Vicente Villardon

References


Examples

```r
# Should be DIRECTLY executable !! ----

categoricalproximities
```

Description

Proximities among individuals using nominal variables.

Usage

CategoricalProximities(Data, SUP = NULL, coefficient = "GOW", transformation = 3, ...)

Arguments

Data  \quad \text{A data frame containing categorical (nominal) variables}
SUP  \quad \text{Supplementary data (Used to project supplementary individuals onto the PCoA configuration, for example)}
coefficient  \quad \text{Similarity coefficient to use (see details)}
transformation  \quad \text{Transformation of the similarity into a distance}
...  \quad \text{Extra parameters}
Details

The function calculates similarities and dissimilarities among a set of objects characterized by a set of nominal variables. The function uses similarities and converts into dissimilarities using a variety of transformations controled by the user.

Value

A list of Values

Author(s)

Jose Luis Vicente Villardon

References


Examples

data(Doctors)
Dis=CategoricalProximities(Doctors, Sup=NULL, coefficient="GOW", transformation=3)
pco=PrincipalCoordinates(Dis)
plot(pco, RowCex=0.7, RowColors=as.integer(Doctors[[1]]), RowLabels=as.character(Doctors[[1]]))

CCA  Canonical Correspondence Analysis

Description

Calculates the solution of a Canonical Correspondence Analysis Biplot

Usage

CCA(P, Z, alpha = 1, dimens = 4)

Arguments

P  Abundance Matrix of sites by species.
Z  Environmental variables measured at the same sites
alpha  Alpha for the biplot decomposition [0,1]. With alpha=1 the emphasis is on the sites and with alpha=0 the emphasis is on the species
dimens  Dimension of the solution
CheckBinaryMatrix

Details

A pair of ecological tables, made of a species abundance matrix and an environmental variables matrix measured at the same sampling sites, is usually analyzed by Canonical Correspondence Analysis (CCA) (Ter Braak, 1986). CCA can be considered as a Correspondence Analysis (CA) in which the ordination axis are constrained to be linear combinations of the environmental variables. Recently the procedure has been extended to other disciplines as Sociology or Psychology and it is potentially useful in many other fields.

Value

A CCA solution object

Author(s)

Jose Luis vicente Villardon

References


Examples

data(riano)
Sp=riano[3:15]
Env=riano[,16:25]
ccabip=CCA(Sp, Env)
plot(ccabip)

CheckBinaryMatrix  Checks if a data matrix is binary

Description

Checks if a data matrix is binary

Usage

CheckBinaryMatrix(x)

Arguments

x       Matrix to check.
CheckBinaryVector

Details
Checks if all the entries of the matrix are either 0 or 1.

Value
TRUE if the matrix is binary.

Author(s)
Jose Luis Vicente-Villardon

Examples

data(spiders)
sp=DataFrame2BinaryMatrix(spiders)
CheckBinaryMatrix(sp)

CheckBinaryVector
Checks if a vector is binary

Description
Checks if all the entries of a vector are 0 or 1.

Usage
CheckBinaryVector(x)

Arguments
x the vector to check

Value
The logical result

Author(s)
Jose Luis Vicente Villardon

Examples
x=c(0, 0, 0, 0, 1, 1, 1, 2)
CheckBinaryVector(x)
Chemical data

Description
Ecological data

Usage
data("Chemical")

Format
A data frame with 324 observations on the following 16 variables.
- Treatment a factor with levels F0N0 F0N1 F0N2 F0N3 F1N0 F1N1 F1N2 F1N3 F2N0 F2N1 F2N2 F2N3 FISH a factor with levels F0 F1 F2
- NUTRIENTS a factor with levels N0 N1 N2 N3
- WEEKS a factor with levels W1 W2 W3 W4 W5 W6 W7 W8 W9
- TEMPERATURE a numeric vector
- pH a numeric vector
- ALKALINITYmeq l a numeric vector
- CO2free a numeric vector
- NH4mg l a numeric vector
- NO3mg l a numeric vector
- SRPmg l P a numeric vector
- TPmg l a numeric vector
- TSSmg l a numeric vector
- CONDUCTIVITYmScm a numeric vector
- TSPmg l P a numeric vector
- Chlorophyllamglm l a numeric vector

Details
Chemical Data

Source
Department of Ecology. University of Leon. (Spain)

References
To add

Examples
data(Chemical)
# may be str(Chemical) ; plot(Chemical) ...
Circle

Draws a circle

Description

Draws a circle for a given radius at the specified center with the given color

Usage

Circle(radius = 1, origin = c(0, 0), color = 1, ...)

Arguments

radius  radius of the circle
origin  Centre of the circle
color   Color of the circle
...     Additional graphical parameters

Details

Draws a circle for a given radius at the specified center with the given color

Value

No value is returned

Author(s)

Jose Luis Vicente Villardon

Examples

plot(0,0)
Circle(1,c(0,0))

ConcEllipse

Concentration ellipse for a set of two-dimensional points

Description

The function calculates a non-parametric concentration ellipse for a set of two-dimensional points.

Usage

ConcEllipse(data, confidence=1, npoints=100)
Arguments

data           The set of two-dimensional points
confidence     Percentage of points to be included in the ellipse
npoints         Number of points to draw the ellipse contour. The hier the number of points the
                smoother is the ellipse.

Details

The procedure uses the Mahalanobis distances to determine the points that will be used for the calculations.

Value

A list with the following fields

  data          Data Used for the calculations
  confidence    The confidence level used
  ellipse       The points on the ellipse contour to be plotted
  center        The center of the points

Author(s)

Jose Luis Vicente Villardon

References


principal components analysis: An empirical study using the balanced bootstrap. Psychological
Methods, 12(3), 359.

Examples

```r
data(iris)
dat=as.matrix(iris[1:50,1:2])
plot(iris[,1], iris[,2],col=iris[,5], asp=1)
E=ConcEllipse(dat, 0.95)
plot(E)
```

Description

Calculates distances among rows of a continuous data matrix or among the rows of two binary matrices.
Usage

ContinuousDistances(x, y = NULL, coef = "Pythagorean", normalizer = "SD", t = 1)

Arguments

x  Main data matrix. Distances among rows are calculated if y=NULL.
y  Supplementary data matrix. If not NULL the distances among the rows of x and y are calculated
coef  Distance coefficient. Use the name or the number (see details)
normalizer  Quantity to normalize the distances
t  Exponent for the Minkowsky

Details

The following coefficients are calculated
1.- Pythagorean = \( \sqrt{\sum((y[i,] - x[j,])^2)/p} \)
2.- Taxonomic = \( \sqrt{\sum(((y[i,] - x[j,])^2)/r^2)/p} \)
3.- City = \( \sum(\text{abs}(y[i,] - x[j,])/r)/p \)
4.- Minkowsky = \( (\sum(\text{abs}(y[i,] - x[j,])/r)^t)/p \)
5.- Divergence = \( \sqrt{\sum((y[i,] - x[j,])^2/(y[i,] + x[j,])^2)/p} \)
6.- dif_sum = \( \sum(\text{abs}(y[i,] - x[j,])/\text{abs}(y[i,] + x[j,]))/p \)
7.- Camberra = \( \sum(\text{abs}(y[i,] - x[j,])/(\text{abs}(y[i,]) + \text{abs}(x[j,]))) \)
8.- Bray_Curtis = \( \sum(\text{abs}(y[i,] - x[j,])/(\sum(y[i,] + x[j,]))) \)
9.- Soergel = \( \sum(\text{abs}(y[i,] - x[j,])/(\sum(\text{apply(rbind(y[i,], x[j,]), 2, max)))) \)
10.- Ware_hedges = \( \sum(\text{abs}(y[i,] - x[j,])/(\sum(\text{apply(rbind(y[i,], x[j,]), 2, max)))) \)

Value

An object of class `proximities`. This has components:

comp1  Description of `comp1`

Author(s)

Jose Luis Vicente-Villardon

References


See Also

PrincipalCoordinates

Examples

data(spiders)
Convert2ThreeWay

Three way array from a two way matrix

Description

Converts a two-dimensional matrix into a list where each cell is the two dimensional data matrix for an occasion or group.

Usage

Convert2ThreeWay(x, groups, columns = TRUE)

Arguments

x The two dimensional matrix

groups A factor defining the groups

columns Are the groups defined for columns?

Details

Converts a two dimensional matrix into a multitable list according to the groups provided by the user. Each field of the list has the name of the corresponding group.

Value

A Multitable list. Each field is the data matrix for a group.

X The multitable list

Author(s)

Jose Luis Vicente Villardon

Examples

data(Chemical)
x= Chemical[,5:16]
X=Convert2ThreeWay(x,Chemical$WEEKS, columns=FALSE)
**ConvertFactors2Integers**

*Convert a factor to integer numbers*

---

**Description**

Convert a factor to integer numbers

**Usage**

ConvertFactors2Integers(x)

**Arguments**

- **x**: A vector with a factor

**Details**

Convert a factor to integer numbers

**Value**

A vector with the converted values

**Author(s)**

Jose Luis Vicente Villardon

**Examples**

```r
# Should be DIRECTLY executable !! ----
```

---

**CrissCross**

*Alternated Least Squares Biplot*

---

**Description**

Alternated Least Squares Biplot with any choice of weights for each element of the data matrix

**Usage**

CrissCross(x, w = matrix(1, dim(x)[1], dim(x)[2]), dimens = 2, a0 = NULL, b0 = NULL, maxiter = 100, tol = 1e-04, addsvd = TRUE, lambda = 0)
Arguments

- **x**: Data Matrix to be analysed
- **w**: Weights matrix. Must be of the same size as X.
- **dimens**: Dimension of the solution.
- **a0**: Starting row coordinates. Random coordinates are calculated if the argument is NULL.
- **b0**: Starting column coordinates. Random coordinates are calculated if the argument is NULL.
- **maxiter**: Maximum number of iterations
- **tol**: Tolerance for the algorithm to converge.
- **addsvd**: Calculate an additional SVD at the end of the algorithm. That makes the final solution more readable
- **lambda**: Constant to add to the diagonal of the matrices to be inverted in order to improve stability when the matrices are ill-conditioned.

Details

The function calculates Alternated Least Squares Biplot with any choice of weights for each element of the data matrix. The function is useful when we want a low rank approximation of a data matrix in which each element of the matrix has a different weight, for example, all the weights are 1 except for the missing elements that are 0, or to model the logarithms of a frequency table using the frequencies as weights.

Value

An object of class .Biplot" with the following components:

- **n**: Number of Rows
- **p**: Number of Columns
- **dim**: Dimension of the Biplot
- **EigenValues**: Eigenvalues
- **Inertia**: Explained variance (Inertia)
- **CumInertia**: Cumulative Explained variance (Inertia)
- **RowCoordinates**: Coordinates for the rows
- **ColCoordinates**: Coordinates for the columns
- **RowContributions**: Contributions for the rows
- **ColContributions**: Contributions for the columns
- **Scale_Factor**: Scale factor for the traditional plot with points and arrows. The row coordinates are multiplied and the column coordinates divided by that scale factor. The look of the plot is better without changing the inner product. For the HJ-Biplot the scale factor is 1.

Author(s)

Jose Luis Vicente Villardon
References


See Also

LogFrequencyBiplot

Examples

data(Protein)
X=as.matrix(Protein[,3:11])
X = InitialTransform(X, transform=5)$X
bip=CrissCross(X)

<table>
<thead>
<tr>
<th>CumSum</th>
<th>Cummulative sums</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

Cummulative sums

Usage

CumSum(X, dimens = 1)

Arguments

X       Data Matrix

dimens   Dimension for summing

Details

Cummulative sums within rows (dimens=1) or columns (dimens=2) of a data matrix

Value

A matrix of the same size as X with cummulative sums within each row or each column

Author(s)

Jose Luis Vicente Villardon

Examples

data(wine)
X=wine[,4:21]
CumSum(X,1)
CumSum(X,2)
Dataframe2BinaryMatrix

Converts a Data Frame into a Binary Data Matrix

Description

Converts a Data Frame into a Binary Data Matrix

Usage

Dataframe2BinaryMatrix(dataf, cuttype = "Median", cut = NULL, BinFact = TRUE)

Arguments

dataf data.frame to be converted
cuttype Type of cut point for continuous variables. Must be "Median" or "Mean". Does not have any effect for factors
cut Personalized cut value for continuous variables.
BinFact Should I treat a factor with two levels as binary. This means that only a single dummy rather than two is used

Details

The function converts a data frame into a Binary Data Matrix (A matrix with entries either 0 or 1).
Factors with two levels are directly transformed into a column of 0/1 entries.
Factors with more than two levels are converted into a binary submatrix with as many rows as n and as many columns as levels or categories. (Indicator matrix)
Integer Variables are treated as factors
Continuous Variables are converted into binary variables using a cut point that can be the median, the mean or a value provided by the user.

Value

A Binary Data Matrix.

Author(s)

Jose Luis Vicente Villardon

Examples

data(spiders)
Dataframe2BinaryMatrix(spiders)
**Dataframe2Matrix4Regression**

*Prepares a matrix for regression from a data frame*

**Description**

Prepares a matrix for regression from a data frame

**Usage**

```r
Dataframe2Matrix4Regression(x, last = TRUE, Intercept = FALSE)
```

**Arguments**

- `x`: A data frame
- `last`: Logical to use the last category of nominal variables as baseline.
- `Intercept`: Logical to tell the function if a constant must be present

**Details**

Nominal variables are converted to a matrix of dummy variables for regression.

**Value**

A matrix ready to use as independent variables in a regression

**Author(s)**

Jose Luis Vicente Vilardon

**Examples**

```
## Should be DIRECTLY executable !! ----
```

---

**DensityBiplot**

*Adds Non-parametric densities to a biplot. Separated densities are calculated for different clusters*

**Description**

Adds Non-parametric densities to a biplot. Separated densities are calculated for different clusters

**Usage**

```r
DensityBiplot(x, y = NULL, grouplabels = NULL, ncontours = 6, groupcolors = NULL, ncolor=20, Color
```
**Arguments**

- **X**: Two dimensional coordinates of the points in a biplot (or any other)
- **y**: A factor containing clusters or groups for separate densities.
- **grouplabels**: Labels for the groups
- **ncontours**: Number of contours to represent on the biplot
- **groupcolors**: Colors for the groups
- **ncolors**: Number of colors for the density patterns
- **ColorType**: One of the following: "1" = rainbow, "2" = heat.colors, "3" = terrain.colors, "4" = topo.colors, "5" = cm.colors

**Details**

Non parametric densities are used to investigate the concentration of row points on different areas of the biplot representation. The densities can be calculated for different groups or clusters in order to investigate if individuals with different characteristics are concentrated on particular areas of the biplot. The procedure is particularly useful with a high number of individuals.

**Value**

No value returned. It has effect on the graph.

**Author(s)**

Jose Luis Vicente Villardon

**References**


**Examples**

```r
bip <- PCA.Biplot(iris[,1:4])
plot(bip, mode="s", CexInd=0.1)
```

---

**Dhats**

*Calculation of Disparities*

**Description**

Calculation of Disparities for a MDS model

**Usage**

```r
Dhats(P, D, W, Model = c("Identity", "Ratio", "Interval", "Ordinal"), Standardize = TRUE)
```
Arguments

P  A matrix of proximities (usually dissimilarities)
D  A matrix of distances obtained from an euclidean configuration
W  A matrix of weights
Model  Measurement level of the proximities
Standardize  Should the Disparities be standardized?

Details

Calculation of disparities using standard or monotone regression depending on the MDS model.

Value

Returns the proximities.

Author(s)

Jose L. Vicente Villardon

References


Examples

```r
## Function is used inside MDS or smacof
```

---

diagonal  

Diagonal matrix from a vector

Description

Creates a diagonal matrix from a vector

Usage

diagonal(d)

Arguments

d  A numerical vector

Value

A diagonal matrix with the values of vector in the diagonal a zeros elsewhere

Author(s)

Jose Luis Vicente Villardon
Examples

dimensionlabels(dimens=3, Root = "Dim")

dimensionlabels(diag(c(1, 2, 3, 4, 5)))

Description

Creates a character vector with labels for the dimensions of the biplot

Usage

DimensionLabels(dimens, Root = "Dim")

Arguments

- dimens: Number of dimensions
- Root: Root of the label

Details

An auxiliary function to create labels for the dimensions. Useful to label the matrices of results

Value

Returns a vector of labels

Author(s)

Jose Luis Vicente Villardon

Examples

DimensionLabels(dimens=3, Root = "Dim")

dlines

Description

Connects two sets of points by lines

Usage

dlines(SetA, SetB, lin = "dotted", color = "black", ...)

Examples

dlines(diag(c(1, 2, 3, 4, 5)))
Arguments

SetA  First set of points
SetB  Second set of points
lin  Line style.
color  Line color
...  Any other graphical parameters

Details

Connects two sets of points by lines

Value

NULL

Author(s)

Based on Graffelman (2013)

References


Examples

## No examples

---

**Doctors**  
*Data set extracted from the Careers of doctorate holders survey carried out by Spanish Statistical Office in 2008.*

Description

The sample data, as part of a large survey, corresponds to 100 people who have the PhD degree and it shows the level of satisfaction of the doctorate holders about some issues.

Usage

data(Doctors)

Format

This data frame contains 100 observation for the following 5 ordinal variables, with four categories each: 1= "Very Satisfied", 2= "Somewhat Satisfied", 3= "Somewhat dissatisfied", 4= "Very dissatisfied")

Salary
Benefits
Job Security
Job Location
Working conditions
EuclideanDistance

Source


Examples

data(Doctors)
## maybe str(Doctors) ; plot(Doctors) ...

EuclideanDistance  Classical Euclidean Distance (Pythagorean Distance)

Description

Calculates the euclidean distances among the rows of an euclidean configurations in any dimensions

Usage

EuclideanDistance(x)

Arguments

x  A matrix containing the euclidean configuration

Details

euclidean distances among the rows of an euclidean configurations in any dimensions

Value

Returns the distance matrix

Author(s)

Jose Luis Vicente Villardon

Examples

x=matrix(runif(20),10,2)
D=EuclideanDistance(x)
**ExpandTable**

Expands a compressed table of patterns and frequencies

**Description**

Expands a compressed table of patterns and frequencies

**Usage**

ExpandTable(table)

**Arguments**

- **table**: A compressed table of patterns and frequencies

**Details**

To simplify the calculations of some of the algorithms we compress the tables by searching for the distinct patterns and its frequencies. This function recovers the original data. It serves also to assign the coordinates on the biplot to the original individuals.

**Value**

A matrix with the original data

**Author(s)**

Jose Luis Vicente Villardon

**Examples**

```r
## Should be DIRECTLY executable !! ----
```

**ExternalBinaryLogisticBiplot**

External Logistic Biplot for binary Data

**Description**

Fits an External Logistic Biplot to the results of a Principal Coordinates Analysis obtained from binary data.

**Usage**

ExternalBinaryLogisticBiplot(Pco, IncludeConst=TRUE, penalization=0.2, freq=NULL, tolerance = 1e-05, maxiter = 100)
Arguments

- **pco**: An object of class "Principal.Coordinates"
- **IncludeConst**: Should the logistic fit include the constant term?
- **penalization**: Penalization for the ridge regression
- **freq**: Frequencies for each observation or pattern (usually 1)
- **tolerance**: Tolerance for convergence
- **maxiter**: Maximum number of iterations

Details

Let $X$ be the matrix of binary data scored as present or absent (1 or 0), in which the rows correspond to $n$ individuals or entries (for example, genotypes) and the columns to $p$ binary characters (for example alleles or bands), let $S = (s_{ij})$ be a matrix containing the similarities among rows, obtained from the binary data matrix, and let $\Delta = (\delta_{ij})$ be the corresponding dissimilarity/distance matrix, taking for example $\delta_{ij} = \sqrt{1 - s_{ij}}$. Despite the fact that, in Cluster Analysis and Principal Coordinates Analysis, interpretation of the variables responsible for grouping or ordination is not straightforward, those methods are normally used to classify individual in which binary variables have been measured. We use a combination of Principal Coordinates Analysis (PCoA), Cluster Analysis (CA) and External Logistic Regression (ELB), as a better way to interpret the binary variables associated to the classification of genotypes. The combination of three standard techniques with some new ideas about the geometry of the procedures, allows to construct a External Logistic Regression (ELB), that helps the interpretation of the variables responsible for the classification or ordination. Suppose we have obtained an euclidean configuration $Y$ obtained from the Principal Coordinates (PCoA) of the similarity matrix. To search for the variables associated to the ordination obtained in PCoA, we can look for the directions in the ordination diagram that better predict the probability of presence of each allele. More formally, if we defined $\pi_{ij} = E(x_{ij}) = \frac{1}{1 + \exp(-b_{j0} + k \sum_{s=1}^{k} b_{js} y_{is})}$ as the expected probability that the allele $j$ be present at genotype for a genotype with coordinates $y_{is}$ ($i=1, \ldots, n; s=1, \ldots, k$) on the ordination diagram, as where $b_{js}$ ($j=1, \ldots, p$) are the logistic regression coefficients that correspond to the $j$th variable (alleles or bands) in the $s$th dimension. The model is a generalized linear model having the logit as a link function. where and $y$'s and $b$'s define a biplot in logit scale. This is called External Logistic Biplot because the coordinates of the genotypes are calculated in an external procedure (PCoA). Given that the $y$'s are known from PCoA, obtaining the $b$'s is equivalent to performing a logistic regression using the $j$-th column of $X$ as a response variable and the columns of $y$ as regressors.

Value

An object of class `External.Binary.Logistic.Biplot` with the fields of the `Principal.Coordinates` object with the following fields added.

- **ColumnParameters**: Parameters resulting from fitting a logistic regression to each column of the original binary data matrix
- **VarInfo**: Information of the fit for each variable
- **VarInfo$Deviances**: A vector with the deviances of each variable calculated as the difference with the null model
- **VarInfo$Dfs**: A vector with degrees of freedom for each variable
ExtractTable

VarInfo$pvalues
A vector with the p values each variable

VarInfo$Nagelkerke
A vector with the Nagelkerke pseudo R-squared for each variable

VarInfo$PercentCorrect
A vector with the percentage of correct classifications for each variable

DevianceTotal
Total Deviance as the difference with the null model

p
p value for the complete representation

TotalPercent
Total percentage of correct classification

Author(s)
Jose Luis Vicente Villardon

References


Examples

data(spiders)
x2=Dataframe2BinaryMatrix(spiders)
colnames(x2)=colnames(spiders)
dist=BinaryProximities(x2)
$pcobip=ExternalBinaryLogisticBiplot(pco)

ExtractTable

Extracts unique patterns and its frequencies for a discrete data matrix (numeric)

Description
Extracts the patterns and the frequencies of a discrete data matrix reducing the size of the data matrix in order to accelerate calculations in some techniques.

Usage
ExtractTable(x)

Arguments
x A matrix of integers containing information of discrete variables. The input matrix must be numerical for the procedure to work properly.
Details

For any numerical matrix, calculates the different patterns and the frequencies associated to each pattern. The result contains the pattern matrix, a vector with the frequencies, a list with rows sharing the same pattern. The final pattern matrix has different ordering than the original matrix.

Value

- **OriginalNames**: Names before grouping the equal rows
- **Patterns**: The reduced table with only unique patterns
- **EqualRows**: A list with as many components as unique patterns specifying the original rows with each pattern. That will allow for the reconstruction of the initial matrix

Author(s)

Jose Luis Vicente-Villardon

Examples

data(spiders)
spidersbin=DataFrame2BinaryMatrix(spiders)
spiderstable=ExtractTable(spidersbin)

---

### FA.Biplot

**Biplot for Factor Analysis.**

Description

Biplot used as a graphical representation of Factor Analysis.

Usage

```r
FA.Biplot(X, dimension = 3, Extraction="PC", Rotation="varimax",
InitComunal="A1", normalize=FALSE, Scores = "Regression",
MethodArgs=NULL, sup.rows = NULL, sup.cols = NULL, ...)
```

Arguments

- **X**: Data Matrix
- **dimension**: Dimension of the solution
- **Extraction**: Method for the extraction of the factors. Can be "PC", "IPF" or "ML" ("Principal Components", "Iterated Principal Factor" or "Maximum Likelihood")
- **Rotation**: Method for the rotation of the factors. Can be "PC", "IPF" or "ML"
- **InitComunal**: Initial communalities for the iterated principal factor method. Can be "A1", "HSC" or "MC" ("All 1", "Highest Simple Correlation" or "Multiple Correlation")
- **normalize**: Should the loadings be normalized
- **Scores**: Method to calculate the Row Scores. Must be "Regression" or "Bartlett".
- **MethodArgs**: Additional arguments associated to the rotation method.
sup.rows    Supplementary or illustrative rows, if any.
sup.cols    Supplementary or illustrative rows, if any.

Details

Biplots represent the rows and columns of a data matrix in reduced dimensions. Usually rows represent individuals, objects or samples and columns are variables measured on them. The most classical versions can be thought as visualizations associated to Principal Components Analysis (PCA) or Factor Analysis (FA) obtained from a Singular Value Decomposition or a related method. From another point of view, Classical Biplots could be obtained from regressions and calibrations that are essentially an alternated least squares algorithm equivalent to an EM-algorithm when data are normal. This routine calculates a biplot as a graphical representation of a classical Factor Analysis allowing for different extraction methods and different rotations.

Value

An object of class "ContinuousBiplot" with the following components:

Title    A general title
Non_Scaled_Data
    Original Data Matrix
Means    Means of the original Variables
Medians  Medians of the original Variables
Deviations  Standard Deviations of the original Variables
Minima   Minima of the original Variables
Maxima   Maxima of the original Variables
P25      25 Percentile of the original Variables
P75      75 Percentile of the original Variables
Gmean    Global mean of the complete matrix
Sup.Rows  Supplementary rows (Non Transformed)
Sup.Cols  Supplementary columns (Non Transformed)
Scaled_Data  Transformed Data
Scaled_Sup.Rows  Supplementary rows (Transformed)
Scaled_Sup.Cols  Supplementary columns (Transformed)
n    Number of Rows
p    Number of Columns
nrowsSup  Number of Supplementary Rows
ncolsSup  Number of Supplementary Columns
dim    Dimension of the Biplot
EigenValues  Eigenvalues
Inertia  Explained variance (Inertia)
CumInertia  Cumulative Explained variance (Inertia)
EV    EigenVectors
Factor2Binary

Structure          Correlations of the Principal Components and the Variables
RowCoordinates    Coordinates for the rows, including the supplementary
ColCoordinates    Coordinates for the columns, including the supplementary
RowContributions  Contributions for the rows, including the supplementary
ColContributions  Contributions for the columns, including the supplementary
Scale_FACTOR      Scale factor for the traditional plot with points and arrows. The row coordinates are multiplied and the column coordinates divided by that scale factor. The look of the plot is better without changing the inner product. For the HJ-Biplot the scale factor is 1.

Author(s)

Jose Luis Vicente Villardon

References


See Also

InitialTransform

Examples

data(Protein)
X=protein[3:11]
bip=FA.Biplot(X, Extraction="ML", Rotation="oblimin")
plot(bip, mode="s", margin=0.05, AddArrow=TRUE)

Factor2Binary

Converts a Factor into its indicator matrix

Description

Converts a factor into a binary matrix with as many columns as categories of the factor

Usage

Factor2Binary(y, Name = NULL)
Arguments

- y: A factor
- Name: Name to use in the final matrix

Value

An indicator binary matrix

Author(s)

Jose Luis Vicente Villardon

Examples

```r
y = factor(c(1, 1, 2, 2, 2, 3, 3, 3, 2, 2, 1, 1, 1))
Factor2Binary(y)
```

Description

Selects a percentage of the data eliminating the observations with higher Mahalanobis distances to the center.

Usage

`Fraction(data, confidence = 1)`

Arguments

- data: Two dimensional data set
- confidence: Percentage to retain. (0-1)

Details

The function is used to select a fraction of the data to be plotted for example when clusters are used. The function eliminates the extreme values.

Value

An object of class `fraction` with the following fields

- data: The original data
- fraction: The selected data
- confidence: The percentage selected

Author(s)

Jose Luis Vicente Villardon
Generalized Procrustes Analysis

Description

Generalized Procrustes Analysis

Usage

GeneralizedProcrustes(x, tolerance = 1e-05, maxiter = 100, Plot = FALSE)

Arguments

x

Three dimensional array with the configurations. The first dimension contains the rows of the configurations, the second contains the columns and the third the number of configurations. So x[i] is the i-th configuration.

tolerance

Tolerance for the Procrustes algorithm.

maxiter

Maximum number of iterations.

Plot

Should the results be plotted?

Details

Generalized Procrustes Analysis for several configurations contained in a three-dimensional array.

Value

An object of class GenProcrustes. This has components:

History

History of Iterations.

X

Initial configurations in a three dimensional array.

RotatedX

Transformed configurations in a three dimensional array.

Scale

Scale factors for each configuration.

Rotations

Rotation Matrices in a three dimensional array.

rss

Residual Sum of Squares.

Fit

Goodness of fit as percent of explained variance.
GetBiplotScales

Author(s)
Jose Luis Vicente-Villardon

References

See Also
PrincipalCoordinates

Examples

data(spiders)
n=dim(spiders)[1]
p=dim(spiders)[2]
prox=array(0,c(n,2,4))

p1=BinaryProximities(spiders,coefficient=5)
prox[,1]=PrincipalCoordinates(p1)$RowCoordinates
p2=BinaryProximities(spiders,coefficient=2)
prox[,2]=PrincipalCoordinates(p2)$RowCoordinates
p3=BinaryProximities(spiders,coefficient=3)
prox[,3]=PrincipalCoordinates(p3)$RowCoordinates
p4=BinaryProximities(spiders,coefficient=4)
prox[,4]=PrincipalCoordinates(p4)$RowCoordinates
GeneralizedProcrustes(prox)

GetBiplotScales  Calculates the scales for the variables on a linear biplot

Description
Calculates the scales for the variables on a linear prediction biplot. There are several types of scales and values that can be shown on the graphical representation. See details.

Usage
GetBiplotScales(Biplot, nticks = 4, TypeScale = "Complete", ValuesScale = "Original")

Arguments
Biplot  Object of class PCA.Biplot
nticks  Number of ticks for the biplot axes
TypeScale  Type of scale to use: "Complete", "StdDev" or "BoxPlot"
ValuesScale  Values to show on the scale: "Original" or "Transformed"
Details

The function calculates the points on the biplot axes where the scales should be placed. There are three types of scales when the transformations of the raw data are made by columns:

"Complete": Covers the whole range of the variable using the number of ticks specified in "nticks". A smaller number of points could be shown if some fall outside the range of the scatter.

"StdDev": The mean +/- 1, 2 and 3 times the standard deviation. A smaller number of points could be shown if some fall outside the range of the scatter.

"BoxPlot": Median, 25, 75 percentiles maximum and minimum values are shown. The extremes of the interquartile range are connected with a thicker line. A smaller number of points could be shown if some fall outside the range of the scatter.

There are two kinds of values that can be shown on the biplot axis:

"Original": The values before transformation. Only makes sense when the transformations are for each column.

"Transformed": The values after transformation, for example, after standardization.

Although the function is public, the end used will not normally use it.

Value

A list with the following components:

- Ticks A list containing the ticks for each variable
- Labels A list containing the labels for each variable

Author(s)

Jose Luis Vicente Villardon

Examples

data(iris)
bip=PCA.Biplot(iris[,1:4])
GetBiplotScales(bip)

GetCCAScales(CCA, nticks = 7, TypeScale = "Complete", ValuesScale = "Original")

Calculates scales for plotting the environmental variables in a Canonical Correspondence Analysis
ginv

Arguments

CCA     A CCA solution object
nticks   Number of ticks to represent
TypeScale Type of scale to represent
ValuesScale Values to represent (Original or Transformed)

Details

Calculates scales for plotting the environmental variables in a Canonical Correspondence Analysis

Value

Returns the points and the labels for each biplot axis

Author(s)

Jose Luis Vicente Villardon

References


Examples

# No examples yet

ginv

G inverse

Description

Calculates the g-inverse of a squared matrix using the eigen decomposition and removing the eigenvalues smaller than a tolerance.

Usage

ginv(X, tol = sqrt(.Machine$double.eps))

Arguments

X     Matrix to calculate the g-inverse
tol   Tolerance.

Details

The function is useful to avoid singularities.
Value
Returns the g-inverse

Author(s)
Jose Luis Vicente Villardon

Examples
data(iris)
x=as.matrix(iris[,1:4])
S= t(x)
ginv(S)

Description
Gower Dissimilarities for mixed types of data

Usage
GowerProximities(x, y=NULL, transformation=3)

Arguments
x Main data. Distances among rows are calculated if y=NULL. Must be a data frame.
y Suplementary data matrix. If not NULL the distances among the rows of x and y are calculated. Must be a data frame with the same columns as x.
transformation Vector with column types. If NULL the data frame types are used.

Details
The transformation \( \sqrt{1-S} \) is applied to the similarity.

Value
An object of class proximities. This has components:

comp1 Description of

Author(s)
Jose Luis Vicente-Villardon

References
Gower Similarities

Examples

data(spiders)

Description

Gower Similarities for mixed types of data

Usage

GowerSimilarities(x, y=NULL, transformation="sqrt(1-S)")

Arguments

x Main data. Distances among rows are calculated if y=NULL. Must be a data frame.
y Suplementary data matrix. If not NULL the distances among the rows of x and y are calculated. Must be a data frame with the same columns as x.
transformation Transformatio to apply to the similarities.

Details

Gower Similarities for mixed types of data. The transformation sqrt(1-S) is applied to the similarity by default.

Value

An object of class proximities. This has components:

comp1 Description of

Author(s)

Jose Luis Vicente-Villardon

References


Examples

data(spiders)
**Hermquad**  
*Gauss-Hermite quadrature*

**Description**
Find the Gauss-Hermite abscissae and weights.

**Usage**

```
Hermquad(N)
```

**Arguments**

- `N`  
  Number of nodes of the quadrature

**Details**
Find the Gauss-Hermite abscissae and weights.

**Value**

- `X`  
  A column vector containing the abscissae.

- `W`  
  A vector containing the corresponding weights.

**Author(s)**
Jose Luis Vicente Villardon (translated from a Matlab function by Greg von Winckel)

**References**

http://www.mathworks.com/matlabcentral/fileexchange/8836-hermite-quadrature/content/hermquad.m

**Examples**

```
Hermquad(10)
```

---

**HJ.Biplot**  
*HJ Biplot with added features.*

**Description**
HJ Biplot with added features.

**Usage**

```
HJ.Biplot(X, dimension = 3, Scaling = 4, sup.rows = NULL, sup.cols = NULL)
```
Arguments

- `X`: Data Matrix
- `dimension`: Dimension of the solution
- `Scaling`: Transformation of the original data. See `InitialTransform` for available transformations.
- `sup.rows`: Supplementary or illustrative rows, if any.
- `sup.cols`: Supplementary or illustrative rows, if any.

Details

Biplots represent the rows and columns of a data matrix in reduced dimensions. Usually rows represent individuals, objects or samples and columns are variables measured on them. The most classical versions can be thought as visualizations associated to Principal Components Analysis (PCA) or Factor Analysis (FA) obtained from a Singular Value Decomposition or a related method. From another point of view, Classical Biplots could be obtained from regressions and calibrations that are essentially an alternated least squares algorithm equivalent to an EM-algorithm when data are normal.

Value

An object of class `ContinuousBiplot` with the following components:

- `title`: A general title
- `Non_Scaled_Data`: Original Data Matrix
- `Means`: Means of the original Variables
- `Medians`: Medians of the original Variables
- `Deviations`: Standard Deviations of the original Variables
- `Minima`: Minima of the original Variables
- `Maxima`: Maxima of the original Variables
- `P25`: 25 Percentile of the original Variables
- `P75`: 75 Percentile of the original Variables
- `Gmean`: Global mean of the complete matrix
- `Sup.Rows`: Supplementary rows (Non Transformed)
- `Sup.Cols`: Supplementary columns (Non Transformed)
- `Scaled_Data`: Transformed Data
- `Scaled_Sup.Rows`: Supplementary rows (Transformed)
- `Scaled_Sup.Cols`: Supplementary columns (Transformed)
- `n`: Number of Rows
- `p`: Number of Columns
- `nrowsSup`: Number of Supplementary Rows
- `ncolsSup`: Number of Supplementary Columns
- `dim`: Dimension of the Biplot
- `EigenValues`: Eigenvalues
InBox

Inertia  Explained variance (Inertia)
CumInertia  Cumulative Explained variance (Inertia)
EV  EigenVectors
Structure  Correlations of the Principal Components and the Variables
RowCoordinates  Coordinates for the rows, including the supplementary
ColCoordinates  Coordinates for the columns, including the supplementary
RowContributions  Contributions for the rows, including the supplementary
ColContributions  Contributions for the columns, including the supplementary
Scale_Factor  Scale factor for the traditional plot with points and arrows. The row coordinates are multiplied and the column coordinates divided by that scale factor. The look of the plot is better without changing the inner product. For the HJ-Biplot the scale factor is 1.

Author(s)
Jose Luis Vicente Villardon

References

See Also
InitialTransform

Examples
```r
## Simple Biplot with arrows
data(Protein)
bip<PCA.Biplot(Protein[,3:11])
plot(bip)

## Biplot with scales on the variables
plot(bip, mode="s", margin=0.2)
```

InBox

Checks if a point is inside a box.

Description
Checks if a point is inside a box. The point is specified by its x and y coordinates and the box with the minimum and maximum values on both coordinate axis: xmin, xmax, ymin, ymax. The vertices of the box are then (xmin, ymin), (xmax, ymin), (xmax, ymax) and (xmin, ymax)

Usage
```
InBox(x, y, xmin, xmax, ymin, ymax)
```
Arguments

- **x**: x coordinate of the point
- **y**: y coordinate of the point
- **xmin**: minimum value of X
- **xmax**: maximum value of X
- **ymin**: minimum value of Y
- **ymax**: maximum value of Y

Value

Returns a logical value: TRUE if the point is inside the box and FALSE otherwise.

Author(s)

Jose Luis Vicente Villardon

Examples

```r
InBox(0, 0, -1, 1, -1, 1)
```

---

**InitialTransform**: *Initial transformation of data*

Description

Initial transformation of data before the construction of a biplot. (or any other technique)

Usage

```r
InitialTransform(X, sup.rows = NULL, sup.cols = NULL, transform = "Standardize columns")
```

Arguments

- **X**: Original Raw Data Matrix
- **sup.rows**: Supplementary or illustrative rows.
- **sup.cols**: Supplementary or illustrative columns.
- **transform**: Transformation to use. See details.

Details

Possible Transformations are:

1. "Raw Data": When no transformation is required.
2. "Subtract the global mean": Eliminate an effect common to all the observations
3. "Double centering": Interaction residuals. When all the elements of the table are comparable. Useful for AMMI models.
4. "Column centering": Remove the column means.
5. "Standardize columns": Remove the column means and divide by its standard deviation.
6. "Row centering": Remove the row means.
7. "Standardize rows": Divide each row by its standard deviation.
8. "Divide by the column means and center": The resulting dispersion is the coefficient of variation.
9. "Normalized residuals from independence" for a contingency table.

The transformation can be provided to the function by using the string between the quotes or just the associated number.

The supplementary rows and columns are not used to calculate the parameters (means, standard deviations, etc). Some of the transformations are not compatible with supplementary data.

**Value**

A list with the following components

- **x**: Transformed data matrix
- **sup.rows**: Transformed supplementary rows
- **sup.cols**: Transformed supplementary columns

**Author(s)**

Jose Luis Vicente Villardon

**References**


**Examples**

data(iris)
x=as.matrix(iris[,1:4])
x=InitialTransform(x, transform=4)
x

---

**Integer2Binary**

*Transforms an Integer Variable into a Binary Variable*

**Description**

Transforms an Integer Variable into a Binary Variable

**Usage**

`Integer2Binary(y, name = "My_Factor")`
**LogFrequencyBiplot**  

**Arguments**

- `y` Vector with the factor name of the factor

**Details**

Transforms an Integer vector into a Binary Indicator Matrix

**Value**

A Binary Data Matrix

**Author(s)**

Jose Luis Vicente-Villardon

**Examples**

```r
dat=c(1, 2, 2, 4, 1, 4, 2, 4)
Integer2Binary(dat,"MyFactor")
```

---

**LogFrequencyBiplot**  

**Weighted Biplot for a table of frequencies**

**Description**

Biplot for the logarithms of the frequencies of a contingency table using the frequencies as weights.

**Usage**

```r
LogFrequencyBiplot(x, Scaling = 1, logoffset = 1, freqoffset = logoffset, ...)
```

**Arguments**

- `x` The frequency table to be biplotted
- `Scaling` Transformation of the matrix after the logarithms
- `logoffset` Constant to add to the frequencies before calculating the logarithms. This is to avoid calculating the logarithm of zero, so, a convenient value for this argument is 1.
- `freqoffset` Constant to add to the frequencies before calculating the weights. This is usually the same as the offset used to add to the frequencies but may be different when we do not want the frequencies zero to influence the biplot, i.e., we want zero weights.
- `...` Any other parameter for the CrissCross procedure.

**Details**

Biplot for the logarithms of the frequencies of a contingency table using the frequencies as weights.
Value

An object of class .Biplot" with the following components:

Title A general title

Non_Scaled_Data

Original Data Matrix

Means Means of the original Variables

Medians Medians of the original Variables

Deviations Standard Deviations of the original Variables

Minima Minima of the original Variables

Maxima Maxima of the original Variables

P25 25 Percentile of the original Variables

P75 75 Percentile of the original Variables

Gmean Global mean of the complete matrix

Sup.Rows Supplementary rows (Non Transformed)

Sup.Cols Supplementary columns (Non Transformed)

Scaled_Data Transformed Data

Scaled_Sup.Rows Supplementary rows (Transformed)

Scaled_Sup.Cols Supplementary columns (Transformed)

n Number of Rows

p Number of Columns

nrowsSup Number of Supplementary Rows

ncolsSup Number of Supplementary Columns

dim Dimension of the Biplot

EigenValues Eigenvalues

Inertia Explained variance (Inertia)

CumInertia Cumulative Explained variance (Inertia)

EV EigenVectors

Structure Correlations of the Principal Components and the Variables

RowCoordinates Coordinates for the rows, including the supplementary

ColCoordinates Coordinates for the columns, including the supplementary

RowContributions Contributions for the rows, including the supplementary

ColContributions Contributions for the columns, including the supplementary

Scale_Factor Scale factor for the traditional plot with points and arrows. The row coordinates are multiplied and the column coordinates divided by that scale factor. The look of the plot is better without changing the inner product. For the HJ-Biplot the scale factor is 1.

Author(s)

Jose Luis Vicente Villardon
logit

References


See Also

CrissCross, ~~~

Examples

data(smoking)
logbip=LogFrequencyBiplot(smoking, Scaling=1, logoffset=0, freqoffset=0)

logit

Logit function

Description

Calculates the logit of a probability

Usage

logit(p)

Arguments

p A probability

Details

Calculates the logit of a probability

Value

The logit of the provided probability

Author(s)

Jose Luis Vicente Villardón
Examples

```r
## Should be DIRECTLY executable !! ----
##-- ==> Define data, use random, 
##--or do help(data=index) for the standard data sets.

# The function is currently defined as
function (p)
{
    logit = log(p/(1 - p))
    return(logit)
}
```

Matrix2Proximities

Description

Converts a matrix of proximities into a Proximities object as used in Principal Coordinates or MDS

Usage

```r
Matrix2Proximities(x, TypeData = "User Provided", Type = c("dissimilarity", "similarity", "product"), Coefficient = NULL, Transformation = NULL, Data = NULL)
```

Arguments

- `x`: The matrix of proximities (a symmetrical matrix)
- `TypeData`: By default is User provided but could be any type.
- `Type`: Type of proximity: dissimilarity, similarity or scalar product. If not provided, the default is dissimilarity
- `Coefficient`: Name of the procedure to calculate the proximities (if any).
- `Transformation`: Transformation used to calculate dissimilarities from similarities (if any)
- `Data`: Raw data used to calculate the proximity (if any).

Details

Converts a matrix of proximities into a Proximities object as used in Principal Coordinates or MDS adding some extra information about the procedure used to obtain the proximities. Is mainly used when the proximities matrix has been provided by the user and not calculated from raw data using BinaryProximities, ContinuousDistances or any other function.

Value

An object of class Proximities containing the proximities matrix and some extra information about it.

Author(s)

Jose Luis Vicente Villardon
**matrixsqrt**

*Matrix squared root*

**Description**

Matrix square root of a matrix using the eigendecomposition.

**Usage**

```r
matrixsqrt(S, tol = sqrt(.Machine$double.eps))
```

**Arguments**

- `S`: A squared matrix
- `tol`: Tolerance for the eigenvalues

**Details**

Matrix square root of a matrix using the eigendecomposition and removing the eigenvalues smaller than a tolerance

**Value**

The matrix square root of the argument

**Author(s)**

Jose Luis Vicente Villardon

**Examples**

```r
data(iris)
x <- matrix(iris[,1:4])
S <- t(x)
matrixsqrt(S)
```
matrixsqrtinv

Inverse of the Matrix squared root

Description

Inverse of the Matrix square root of a matrix using the eigendecomposition.

Usage

matrixsqrtinv(S, tol = sqrt(.Machine$double.eps))

Arguments

S A squared matrix
tol Tolerance for the eigenvalues

Details

Inverse of the Matrix square root of a matrix using the eigendecomposition and removing the eigenvalues smaller than a tolerance

Value

The inverse matrix square root of the argument

Author(s)

Jose Luis Vicente Villardon

See Also

ginv

Examples

data(iris)
x=as.matrix(iris[,1:4])
S=t(x)
matrixsqrtinv(S)
Multidimensional Scaling

Description

Multidimensional Scaling using SMACOF algorithm and Bootstraping the coordinates.

Usage

```r
MDS(Proximities, W = NULL, Model = c("Identity", "Ratio", "Interval", "Ordinal"), dimsol = 2, 
maxiter = 100, maxerror = 1e-06, Bootstrap = FALSE, nB = 200, ProcrustesRot = TRUE, 
BootstrapMethod = c("Sampling", "Permutation"), 
StandardizeDisparities = FALSE, ShowIter = FALSE)
```

Arguments

- `Proximities` An object of class proximities
- `W` A matrix of weights
- `Model` MDS model.
- `dmsol` Dimension of the solution
- `maxiter` Maximum number of iterations of the algorithm
- `maxerror` Tolerance for convergence of the algorithm
- `Bootstrap` Should Bootstraping be performed?
- `nB` Number of Bootstrap samples.
- `ProcrustesRot` Should the bootstrap replicates be rotated to match the initial configuration using Procrustes?
- `BootstrapMethod` The bootstrap is performed by sampling or permuting the residuals?
- `StandardizeDisparities` Should the disparities be standardized
- `ShowIter` Show the iteration process

Details

Multidimensional Scaling using SMACOF algorithm and Bootstraping the coordinates. MDS performs multidimensional scaling of proximity data to find a least-squares representation of the objects in a low-dimensional space. A majorization algorithm guarantees monotone convergence for optionally transformed, metric and nonmetric data under a variety of models.

Value

An object of class `PrincipalCoordinates` and `MDS`. The function adds the information of the MDS to the object of class `proximities`. Together with the information about the proximities the object has:

- `Analysis` The type of analysis performed, "MDS" in this case
- `Model` MDS model used
- `RowCoordinates` Coordinates for the objects in the MDS procedure
RawStress Raw Stress values
stress1 stress formula 1
stress2 stress formula 2
ssstress1 ssstress formula 1
ssstress2 ssstress formula 2
rsq Squared correlation between disparities and distances
Spearman Spearman correlation between disparities and distances
Kendall Kendall correlation between disparities and distances
BootstrapInfo The result of the bootstrap calculations

Author(s)

Jose Luis Vicente Villardon

References


See Also

BootstrapSmacof

Examples

data(spiders)
Dis=BinaryProximities(spiders)
MDSsol=MDS(Dis, Bootstrap=FALSE)
plot(MDSsol)
Description

Model based clustering using mixtures of gaussian distributions.

Usage

\[
\text{MGC}(x, \text{NG} = 2, \text{init} = "km", \text{RemoveOutliers}=\text{FALSE}, \text{ConfidOutliers}=0.995, \text{tolerance} = 1e-07, \text{maxiter} = \ldots)
\]

Arguments

- \(x\): The data matrix
- \(\text{NG}\): Number of groups or clusters to obtain
- \(\text{init}\): Initial centers can be obtained from k-means ("km") or at random ("rd")
- \(\text{RemoveOutliers}\): Should the extreme values be removed to calculate the clusters?
- \(\text{ConfidOutliers}\): Percentage of the points to keep for the calculations when RemoveOutliers is true.
- \(\text{tolerance}\): Tolerance for convergence
- \(\text{maxiter}\): Maximum number of iterations
- \(\text{show}\): Should the likelihood at each iteration be shown?
- \(\ldots\): Maximum number of iterations Any other parameter that can affect k-means if that is the initial configuration

Details

A basic algorithm for clustering with mixtures of gaussians with no restrictions on the covariance matrices

Value

Clusters

Author(s)

Jose Luis Vicente Villardon

References

Me falta

Examples

\[
x = \text{as.matrix(iris[,1:4])}
\text{mod1} = \text{MGC}(x, \text{NG}=3)
\text{plot(iris[,1:4], col=mod1\$Classification)}
\text{table(iris[,5], mod1\$Classification)}
\]
MonotoneRegression

**Description**

Performs weighted isotonic (monotone) regression using the non-negative weights in w. The function is a direct translation of the matlab function lsqisotonic.

**Usage**

`MonotoneRegression(x, y, w = NULL)`

**Arguments**

- **x**: The independent variable vector
- **y**: The dependent variable vector
- **w**: A vector of weights

**Details**

YHAT = MonotoneRegression(X,Y) returns a vector of values that minimize the sum of squares (Y - YHAT).^2 under the monotonicity constraint that X(I) > X(J) => YHAT(I) >= YHAT(J), i.e., the values in YHAT are monotonically non-decreasing with respect to X (sometimes referred to as "weak monotonicity"). LSQISOTONIC uses the "pool adjacent violators" algorithm.

If X(I) == X(J), then YHAT(I) may be <, ==, or > YHAT(J) (sometimes referred to as the "primary approach"). If ties do occur in X, a plot of YHAT vs. X may appear to be non-monotonic at those points. In fact, the above monotonicity constraint is not violated, and a reordering within each group of ties, by ascending YHAT, will produce the desired appearance in the plot.

**Value**

The fitted values after the monotone regression

**Note**

The function is a direct translation of the matlab function lsqisotonic.

**Author(s)**

Jose L. Vicente Villardon (from a matlab function)

**References**


**Examples**

```r
## Used inside MDS
```
Description

Moth data

Usage

data("moth")

Format

A data frame with 12 observations on the following 14 variables.

s1  a numeric vector
s2  a numeric vector
s3  a numeric vector
s4  a numeric vector
s5  a numeric vector
s6  a numeric vector
s7  a numeric vector
s8  a numeric vector
s9  a numeric vector
s10 a numeric vector
s11 a numeric vector
s12 a numeric vector
s13 a numeric vector
s14 a numeric vector

Details

Moth data

Source

Withaker

References

Application of the Parametric Bootstrap to Models that Incorporate a Singular Value Decomposition

Examples

data(moth)
## maybe str(moth); plot(moth) ...
MultiTableStatistics

MultiTableStatistics

Statistics for multiple tables

Description

Statistics for multiple tables

Usage

MultiTableStatistics(X)

Arguments

X A multiple table
**Details**

Statistics for multiple tables

**Value**

A list with vectors of statistics for each table

**Author(s)**

Jose Luis Vicente Villardon

**Examples**

```c
### Should be DIRECTLY executable !! ----
```

---

**Description**

Initial Transformation of a multi table object

**Usage**

```r
MultitableTransform(X, InitTransform = "Standardize columns")
```

**Arguments**

- `X` Multi-table object
- `InitTransform` Initial Transformation

**Details**

Initial Transformation of a multi table object

**Value**

The table transformed

**Author(s)**

Jose Luis Vicente Villardon

**Examples**

```c
### Should be DIRECTLY executable !! ----
```
### NiceNumber

#### Description

Calculates a close nice number, i.e. a number with simple decimals.

#### Usage

`NiceNumber(x = 6, round = TRUE)`

#### Arguments

- `x` A number
- `round` Should the number be rounded?

#### Details

Calculates a close nice number, i.e. a number with simple decimals.

#### Value

A number with simple decimals

#### Author(s)

Jose Luis Vicente Villardon

#### References


#### See Also

PrettyTicks

#### Examples

`NiceNumber(0.892345)`
NominalDistances

Distances among individuals with nominal variables

Description

This function computes several measures of distance (or similarity) among individuals from a nominal data matrix.

Usage

NominalDistances(x, method = 1, diag = FALSE, upper = FALSE, similarity = TRUE)

Arguments

x Matrix or data.frame with the nominal variables.
method An integer between 1 and 6. See details
diag A logical value indicating whether the diagonal of the distance matrix should be printed.
upper a logical value indicating whether the upper triangle of the distance matrix should be printed.
similarity A logical value indicating whether the similarity matrix should be computed.

Details

Let be the table of nominal data. All these distances are of type \( d = \sqrt{1-s} \) with \( s \) a similarity coefficient.

1 = Overlap method The overlap measure simply counts the number of attributes that match in the two data instances.

2 = Eskin Eskin et al. proposed a normalization kernel for record-based network intrusion detection data. The original measure is distance-based and assigns a weight of \( \frac{2}{n^2} \) for mismatches; when adapted to similarity, this becomes a weight of \( \frac{n^2}{n^2+2} \). This measure gives more weight to mismatches that occur on attributes that take many values.

3 = IOF (Inverse Occurrence Frequency) This measure assigns lower similarity to mismatches on more frequent values. The IOF measure is related to the concept of inverse document frequency which comes from information retrieval, where it is used to signify the relative number of documents that contain a specific word.

4 = OF (Occurrence Frequency) This measure gives the opposite weighting of the IOF measure for mismatches, i.e., mismatches on less frequent values are assigned lower similarity and mismatches on more frequent values are assigned higher similarity.

5 = Goodall3 This measure assigns a high similarity if the matching values are infrequent regardless of the frequencies of the other values.

6 = Lin This measure gives higher weight to matches on frequent values, and lower weight to mismatches on infrequent values.

Value

An object of class distance
Author(s)
Jose L. Vicente-Villardon

References

See Also
BinaryDistances,ContinuousDistances

Examples
## Not run:
data(Env)
Distance<-NominalDistances(Env,upper=TRUE,diag=TRUE,similarity=FALSE,method=1)

## End(Not run)

——

Numeric2Binary  Converts a numeric variable into a binary one

Description
Converts a numeric variable into a binary one using a cut point

Usage
Numeric2Binary(y, name= "MyVar", cut = NULL)

Arguments
y  Vector containing the numeric values
name  Name of the variable
cut  Cut point to cut the values of the variable. If is NULL the median is used.

Details
Converts a numeric variable into a binary one using a cut point. If the cut is NULL the median is used.

Value
A binary Variable

Author(s)
Jose Luis Vicente-Villardon
ones

See Also

Dataframe2BinaryMatrix

Examples

```r
y = c(1, 1.2, 3.2, 2.4, 1.7, 2.2, 2.7, 3.1)
Numeric2Binary(y)
```

Description

Square matrix of ones

Usage

```r
ones(n)
```

Arguments

- `n`: Order of the matrix

Details

Square matrix of ones

Value

A matrix of ones of order n.

Author(s)

Jose Luis Vicente Villardon

Examples

```r
ones(5)
```
OrdinalLogisticFit

Fits an ordinal logistic regression with ridge penalization

Description

This function fits a logistic regression between a dependent ordinal variable $y$ and some independent variables $x$, and solves the separation problem using ridge penalization.

Usage

OrdinalLogisticFit($y$, $x$, penalization = 0.1, tol = 1e-04, maxiter = 200, show = FALSE)

Arguments

$y$  Dependent variable.

$x$  A matrix with the independent variables.

penalization  Penalization used to avoid singularities.

tol  Tolerance for the iterations.

maxiter  Maximum number of iterations.

show  Should the iteration history be printed?

Details

The problem of the existence of the estimators in logistic regression can be seen in Albert (1984); a solution for the binary case, based on the Firth’s method, Firth (1993) is proposed by Heinze(2002). All the procedures were initially developed to remove the bias but work well to avoid the problem of separation. Here we have chosen a simpler solution based on ridge estimators for logistic regression Cessie(1992).

Rather than maximizing $L_j(G | b_{j0}, B_j)$ we maximize

$$L_j(G | b_{j0}, B_j) - \lambda (\|b_{j0}\| + \|B_j\|)$$

Changing the values of $\lambda$ we obtain slightly different solutions not affected by the separation problem.

Value

An object of class "pordlogist". This has components:

nobs  Number of observations

J  Maximum value of the dependent variable

nvar  Number of independent variables

fitted.values  Matrix with the fitted probabilities

pred  Predicted values for each item

Covariances  Covariances matrix

clasif  Matrix of classification of the items

PercentClasif  Percent of good classifications
coefficients  Estimated coefficients for the ordinal logistic regression
thresholds  Thresholds of the estimated model
logLik  Logarithm of the likelihood
penalization  Penalization used to avoid singularities
Deviance  Deviance of the model
DevianceNull  Deviance of the null model
Dif  Difference between the two deviances values calculated
df  Degrees of freedom
pval  p-value of the contrast
CoxSnell  Cox-Snell pseudo R squared
Nagelkerke  Nagelkerke pseudo R squared
MacFaden  Nagelkerke pseudo R squared
iter  Number of iterations made

Author(s)
Jose Luis Vicente-Villardon

References

Examples
# No examples yet

OrdLogBipEM  Alternated EM algorithm for Ordinal Logistic Biplots

Description
This function computes, with an alternated algorithm, the row and column parameters of an Ordinal Logistic Biplot for ordered polytomous data. The row coordinates (E-step) are computed using multidimensional Gauss-Hermite quadratures and Expected a posteriori (EAP) scores and parameters for each variable or items (M-step) using Ridge Ordinal Logistic Regression to solve the separation problem present when the points for different categories of a variable are completely separated on the representation plane and the usual fitting methods do not converge. The separation problem is present in almost every data set for which the goodness of fit is high.
Usage

**OrdLogBipEM**

OrdLogBipEM(\texttt{Data, freq=NULL, dim = 2, nnodes = 15, tol = 0.0001, maxiter = 100, maxiterlogist = 100, penalization = 0.2, show = FALSE, initial = 1, alfa = 1, Orthogonalize=TRUE, Varimax=TRUE, ...})

Arguments

- **Data**: Data frame with the ordinal data. All the variables must be ordered factors.
- **freq**: Frequencies for compacted tables
- **dim**: Dimension of the solution
- **nnodes**: Number of nodes for the multidimensional Gauss-Hermite quadrature
- **tol**: Value to stop the process of iterations.
- **maxiter**: Maximum number of iterations for the biplot procedure.
- **maxiterlogist**: Maximum number of iterations for the logistic regression step or the Mirt initial configuration.
- **penalization**: Penalization used in the diagonal matrix to avoid singularities.
- **show**: Boolean parameter to specify if the user wants to see every iteration.
- **initial**: Method used to choose the initial ability in the algorithm. Default value is 1.
- **alfa**: Optional parameter to calculate row and column coordinates in Simple correspondence analysis if the initial parameter is equal to 1.
- **Orthogonalize**: Should the final row coordinates be orthogonalized?. The column parameters have to be recalculated.
- **Varimax**: Should the final row coordinates be rotated using the varimax procedure?.
- **...**: Additional arguments for mirt.

Value

An object of class "**Ordinal.Logistic.Biplot**". This has components:

- **RowCoordinates**: Coordinates for the rows or the individuals
- **ColumnParameters**: List with information about the Ordinal Logistic Models calculated for each variable including: estimated parameters with thresholds, percents of correct classifications, and pseudo-R Squared
- **loadings**: Factor loadings
- **LogLikelihood**: Logarithm of the likelihood
- **r2**: R squared coefficient
- **Ncats**: Number of the categories of each variable

Author(s)

Jose Luis Vicente-Villardon

References

**Examples**

```r
## Not run:
data(Doctors)
olb = OrdLogBipEM(Doctors, dim = 2, nnodes = 10, initial=4, tol = 0.001, maxiter = 100, penalization = 0.1, olb
summary(olb)
PlotOrdinalResponses(olb)
## End(Not run)
```

**Description**

Plots an ordinal variable on the biplot from its fitted parameters.

**Usage**

```r
OrdVarBiplot(bi1, bi2, threshold, xmin = -3, xmax = 3, ymin = -3, ymax = 3, label = "Point", mode =
```

**Arguments**

- `bi1`: Slope for the first dimension to plot
- `bi2`: Slope for the second dimension to plot
- `threshold`: Thresholds for each category of the variable
- `xmin`: Minimum value of the X on the plot
- `xmax`: Maximum value of the X on the plot
- `ymin`: Minimum value of the Y on the plot
- `ymax`: Maximum value of the X on the plot
- `label`: Label of the variable
- `mode`: Mode of the plot (as in a regular biplot)
- `cexMarks`: Size of the tick marks
- `cexPoint`: Size of the point
- `pchPoint`: Mark for the point
- `Color`: Color
- `tl`: Tick Length
- `textpos`: Position of the label
- `...`: Any additional graphical parameter

**Details**

Plots an ordinal variable on the biplot from its fitted parameters. The plot uses the same parameters as any other biplot.

**Value**

Returns a graphical representation of the ordinal variable on the current plot.
Author(s)
Jose Luis Vicente Villardon

References

Examples
### Should be DIRECTLY executable !! ----

OrdVarCoordinates

Coordinates of an ordinal variable on the biplot.

Description
Coordinates of an ordinal variable on the biplot.

Usage
OrdVarCoordinates(tr, b = c(1, 1), inf = -12, sup = 12, step = 0.01, plotresponse = FALSE, label = "Item", labx = "z", laby = "Probability", catnames = NULL, Legend = TRUE, LegendPos = 1)

Arguments
- tr: A vector containing the thresholds of the model, that is, the constant for each category of the ordinal variable
- b: Vector containing the common slopes for all categories of the ordinal variable
- inf: The inferior limit of the values to be sampled on the biplot axis (it depends on the scale of the biplot).
- sup: The superior limit of the values to be sampled on the biplot axis (it depends on the scale of the biplot).
- step: Increment (step) of the sequence
- plotresponse: Should the item be plotted
- label: Label of the item.
- labx: Label for the X axis in the summary of the item.
- laby: Label for the Y axis in the summary of the item.
- catnames: Names of the categories.
- Legend: Should a legend be plotted
- LegendPos: Position of the legend.

Details
The function calculates the coordinates of the points that define the separation among the categories of an ordinal variable projected onto an ordinal logistic biplot.
Value

An object of class OrdVarCoord

- `z`: Values of the cut points on the scale of the biplot axis (not used)
- `points`: The points for the marks to be represented on the biplot.
- `labels`: The labels for the points
- `hidden`: Are there any hidden categories? (Categories whose probability is never higher than the probabilities of the rest)
- `cathidden`: Number of the hidden categories

Author(s)

Jose Luis Vicente Villardon

References


Examples

```r
# No examples
```

OrthogonalizeScores  Orthogonalize a set of Scores calculated by other procedure

Description

Orthogonalize a set of Scores calculated by other procedure

Usage

```r
OrthogonalizeScores(scores)
```

Arguments

- `scores`: A matrix containing the scores

Details

Orthogonalize a set of Scores calculated by other procedure projecting onto the dimensions defined by the eigenvectors of the covariance matrix

Value

The orthogonalised scores.

Author(s)

Jose Luis Vicente Villardon
Examples

```#
--- Should be DIRECTLY executable !! ----
```

## PCA.Biplot

### Classical PCA Biplot with added features.

#### Description

Classical PCA Biplot with added features.

#### Usage

```r
PCA.Biplot(X, alpha = 1, dimension = 3, Scaling = 4, sup.rows = NULL, sup.cols = NULL)
```

#### Arguments

- `X` Data Matrix
- `alpha` A number between 0 and 1. 0 for GH-Biplot, 1 for JK-Biplot and 0.5 for SQRT-Biplot. Use 2 or any other value not in the interval [0,1] for HJ-Biplot.
- `dimension` Dimension of the solution
- `Scaling` Transformation of the original data. See InitialTransform for available transformations.
- `sup.rows` Supplementary or illustrative rows, if any.
- `sup.cols` Supplementary or illustrative rows, if any.

#### Details

Biplots represent the rows and columns of a data matrix in reduced dimensions. Usually rows represent individuals, objects or samples and columns are variables measured on them. The most classical versions can be thought as visualizations associated to Principal Components Analysis (PCA) or Factor Analysis (FA) obtained from a Singular Value Decomposition or a related method. From another point of view, Classical Biplots could be obtained from regressions and calibrations that are essentially an alternated least squares algorithm equivalent to an EM-algorithm when data are normal.

#### Value

An object of class ContinuousBiplot with the following components:

- **Title** A general title
- **Non_Scaled_Data** Original Data Matrix
- **Means** Means of the original Variables
- **Medians** Medians of the original Variables
- **Deviations** Standard Deviations of the original Variables
- **Minima** Minima of the original Variables
- **Maxima** Maxima of the original Variables
### PCA.Biplot

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P25</td>
<td>25 Percentile of the original Variables</td>
</tr>
<tr>
<td>P75</td>
<td>75 Percentile of the original Variables</td>
</tr>
<tr>
<td>Gmean</td>
<td>Global mean of the complete matrix</td>
</tr>
<tr>
<td>Sup.Rows</td>
<td>Supplementary rows (Non Transformed)</td>
</tr>
<tr>
<td>Sup.Cols</td>
<td>Supplementary columns (Non Transformed)</td>
</tr>
<tr>
<td>Scaled_Data</td>
<td>Transformed Data</td>
</tr>
<tr>
<td>Scaled_Sup.Rows</td>
<td>Supplementary rows (Transformed)</td>
</tr>
<tr>
<td>Scaled_Sup.Cols</td>
<td>Supplementary columns (Transformed)</td>
</tr>
<tr>
<td>n</td>
<td>Number of Rows</td>
</tr>
<tr>
<td>p</td>
<td>Number of Columns</td>
</tr>
<tr>
<td>nrowsSup</td>
<td>Number of Supplementary Rows</td>
</tr>
<tr>
<td>ncolsSup</td>
<td>Number of Supplementary Columns</td>
</tr>
<tr>
<td>dim</td>
<td>Dimension of the Biplot</td>
</tr>
<tr>
<td>EigenValues</td>
<td>Eigenvalues</td>
</tr>
<tr>
<td>Inertia</td>
<td>Explained variance (Inertia)</td>
</tr>
<tr>
<td>CumInertia</td>
<td>Cumulative Explained variance (Inertia)</td>
</tr>
<tr>
<td>EV</td>
<td>EigenVectors</td>
</tr>
<tr>
<td>Structure</td>
<td>Correlations of the Principal Components and the Variables</td>
</tr>
<tr>
<td>RowCoordinates</td>
<td>Coordinates for the rows, including the supplementary</td>
</tr>
<tr>
<td>ColCoordinates</td>
<td>Coordinates for the columns, including the supplementary</td>
</tr>
<tr>
<td>RowContributions</td>
<td>Contributions for the rows, including the supplementary</td>
</tr>
<tr>
<td>ColContributions</td>
<td>Contributions for the columns, including the supplementary</td>
</tr>
<tr>
<td>Scale_Factor</td>
<td>Scale factor for the traditional plot with points and arrows. The row coordinates are multiplied and the column coordinates divided by that scale factor. The look of the plot is better without changing the inner product. For the HJ-Biplot the scale factor is 1.</td>
</tr>
</tbody>
</table>

### Author(s)

Jose Luis Vicente Villardon

### References


See Also

InitialTransform

Examples

```r
## Simple Biplot with arrows
data(Protein)
bip=PCA.Biplot(Protein[,3:11])
plot(bip)

## Biplot with scales on the variables
plot(bip, mode="s", margin=0.2)
```

plot.CA.sol  
*Plot the solution of a Correspondence Analysis*

Description

Plots the solution of a Correspondence Analysis

Usage

```r
## S3 method for class 'CA.sol'
plot(x, ...)
```

Arguments

- `x`  
  A CA.sol object
- `...`  
  Any other biplot and graphical parameters

Details

Plots the solution of a Correspondence Analysis

Value

No value returned

Author(s)

Jose Luis Vicente Villardon

References

Add some references here
plot.Canonical.Biplot

See Also

plot.ContinuousBiplot

Examples

data(riano)
Sp=riano[,3:15]
cabip=CA(Sp)
plot(cabip)

plot.Canonical.Biplot  Plots a Canonical Biplot

Description

Plots a Canonical Biplot

Usage

## S3 method for class 'Canonical.Biplot'
plot(x, A1 = 1, A2 = 2, ScaleGraph = TRUE, PlotGroups =
    TRUE, PlotVars = TRUE, PlotInd = TRUE, LabelInd =
    TRUE, CexGroup = 1, PchGroup = 16, margin = 0.1,
    AddLegend = TRUE, ShowAxes = FALSE, LabelAxes = TRUE,
    LabelGroups = TRUE, PlotCircle = TRUE, ConvexHulls =
    FALSE, TypeCircle = "M", ColorGroups = NULL, ColorVars
    = NULL, LegendPos = "topright", ColorInd = NULL,
    voronoi = TRUE, mode = "a", TypeScale = "Complete",
    ValuesScale = "Original", MinQualityVars = 0, dpg = 0,
    dpi = 0, PredPoints = 0, PlotAxis = FALSE, CexInd =
    NULL, CexVar = NULL, PchInd = NULL, PchVar = NULL,
    ColorVar = NULL, ShowAxis = FALSE, VoronioColor =
    "black", ShowBox=TRUE, ...)

Arguments

x        An object of class "Canonical.Biplot"
A1       Dimension for the first axis. 1 is the default.
A2       Dimension for the second axis. 2 is the default.
ScaleGraph  Reescale the coordinates to optimal matching.
PlotGroups  Shoud the group centers be plotted?
PlotVars   Should the variables be plotted?
PlotInd    Should the individuals be plotted?
LabelInd   Should the individuals be labeled?
CexGroup   Sizes of the points for the groups
PchGroup   Markers for the group
margin     margin for the graph
AddLegend  Should a legend with the groups be added?
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShowAxes</td>
<td>Should outside axes be shown?</td>
</tr>
<tr>
<td>LabelAxes</td>
<td>Should outside axes be labelled?</td>
</tr>
<tr>
<td>LabelGroups</td>
<td>Should the groups be labeled?</td>
</tr>
<tr>
<td>PlotCircle</td>
<td>Should the confidence regions for the groups be plotted?</td>
</tr>
<tr>
<td>ConvexHulls</td>
<td>Should the convex hulls containing the individuals for each group be plotted?</td>
</tr>
<tr>
<td>TypeCircle</td>
<td>Type of confidence region: Univariate (U), Bonferroni(B), Multivariate (M) or Classical (C)</td>
</tr>
<tr>
<td>ColorGroups</td>
<td>User colors for the groups. Default colors will be used if NULL.</td>
</tr>
<tr>
<td>ColorVars</td>
<td>User colors for the variables. Default colors will be used if NULL.</td>
</tr>
<tr>
<td>LegendPos</td>
<td>Position of the legend.</td>
</tr>
<tr>
<td>ColorInd</td>
<td>User colors for the individuals. Default colors will be used if NULL.</td>
</tr>
<tr>
<td>voronoi</td>
<td>Should the voronoi diagram with the prediction regions for each group be plotted?</td>
</tr>
<tr>
<td>mode</td>
<td>Mode of the biplot: &quot;p&quot;, &quot;a&quot;, &quot;b&quot;, &quot;h&quot;, &quot;ah&quot; and &quot;s&quot;.</td>
</tr>
<tr>
<td>TypeScale</td>
<td>Type of scale to use: &quot;Complete&quot;, &quot;StdDev&quot; or &quot;BoxPlot&quot;</td>
</tr>
<tr>
<td>ValuesScale</td>
<td>Values to show on the scale: &quot;Original&quot; or &quot;Transformed&quot;</td>
</tr>
<tr>
<td>MinQualityVars</td>
<td>Minimum quality of representation for a variable to be plotted</td>
</tr>
<tr>
<td>dpdg</td>
<td>A set of indices with the variables that will show the projections of the groups</td>
</tr>
<tr>
<td>dpi</td>
<td>A set of indices with the variables that will show the projections of the individuals</td>
</tr>
<tr>
<td>PredPoints</td>
<td>A vector with integers. The group centers listed in the vector are projected onto all the variables.</td>
</tr>
<tr>
<td>PlotAxis</td>
<td>Not Used</td>
</tr>
<tr>
<td>CexInd</td>
<td>Size of the points for individuals.</td>
</tr>
<tr>
<td>CexVar</td>
<td>Size of the points for variables.</td>
</tr>
<tr>
<td>PchInd</td>
<td>Markers of the points for individuals.</td>
</tr>
<tr>
<td>PchVar</td>
<td>Markers of the points for variables.</td>
</tr>
<tr>
<td>ColorVar</td>
<td>Colors of the points for variables.</td>
</tr>
<tr>
<td>ShowAxis</td>
<td>Should axis scales be shown?</td>
</tr>
<tr>
<td>VoronoiColor</td>
<td>Color for the Voronoi diagram</td>
</tr>
<tr>
<td>ShowBox</td>
<td>Should a box around the points be plotted?</td>
</tr>
<tr>
<td>...</td>
<td>Any other graphical parameters</td>
</tr>
</tbody>
</table>

**Details**

The function plots the results of a Canonical Biplot. The coordinates for Groups, Individuals and Variables can be shown or not on the plot, each of the three can also be labeled separately. The are parameters to control the way each different set of coordinates is plotted and labeled.

There are several modes for plotting the biplot.

"p".- Points (Rows and Columns are represented by points)

"a".- Arrows (The traditional representation with points for rows and arrows for columns)

"b".- The arrows for the columns are extended to both extremes of the plot and labeled outside the plot area.
"h": - The arrows for the columns are extended to the positive extreme of the plot and labeled outside the plot area.

"ah": - Same as arrows but labeled outside the plot area.

"s": - The directions (or biplot axes) have a graded scale for prediction of the original values.

The `TypeScale` argument applies only to the "s" mode. There are three types:

"Complete": - An equally spaced scale covering the whole range of the data is calculated.

"StdDev": - Mean with one, two and three standard deviations

"BoxPlot": - Box-Plot like Scale (Median, 25 and 75 percentiles, maximum and minimum values.)

The `ValuesScale` argument applies only to the "s" mode and controls if the labels show the Original or Transformed values.

Some of the initial transformations are not compatible with some of the types of biplots and scales. For example, it is not possible to recover by projection the original values when you double centre de data. In that case you have the residuals for interaction and only the transformed values make sense.

**Value**

No value returned

**Author(s)**

Jose Luis Vicente Villardon

**References**


**Examples**

```r
data(wine)
x=as.numeric(wine[,4:21])
canbip=CanonicalBiplot(x, group=wine$Group)
plot(canbip, TypeCircle="U")
```
plot.CCA.sol  
Plots the solution of a Canonical Correspondence Analysis

Description

Plots the solution of a Canonical Correspondence Analysis using similar parameters to the continuous biplot.

Usage

```r
## S3 method for class 'CCA.sol'
plot(x, A1 = 1, A2 = 2, ShowAxis = FALSE, margin = 0,
     PlotSites = TRUE, PlotSpecies = TRUE, PlotEnv = TRUE,
     LabelSites = TRUE, LabelSpecies = TRUE, LabelEnv = TRUE,
     TypeSites = "wa", SpeciesQuality = FALSE,
     MinQualityVars = 0.3, dp = 0, pr = 0, PlotAxis = FALSE,
     TypeScale = "Complete", ValuesScale = "Original",
     mode = "a", CexSites = NULL, CexSpecies = NULL,
     CexVar = NULL, ColorSites = NULL, ColorSpecies = NULL,
     ColorVar = NULL, PchSites = NULL, PchSpecies = NULL,
     PchVar = NULL, SizeQualSites = FALSE,
     SizeQualSpecies = FALSE, SizeQualVars = FALSE,
     ColorQualSites = FALSE, ColorQualSpecies = FALSE,
     ColorQualVars = FALSE, SmartLabels = FALSE, ...)
```

Arguments

- `x`
- `A1`
- `A2`
- `ShowAxis`
- `margin`
- `PlotSites`
- `PlotSpecies`
- `PlotEnv`
- `LabelSites`
- `LabelSpecies`
- `LabelEnv`
- `TypeSites`
- `SpeciesQuality`
- `MinQualityVars`
- `dp`
- `pr`
- `PlotAxis`
- `TypeScale`
ValuesScale
mode
CexSites
CexSpecies
CexVar
ColorSites
ColorSpecies
ColorVar
PchSites
PchSpecies
PchVar
SizeQualSites
SizeQualSpecies
SizeQualVars
ColorQualSites
ColorQualSpecies
ColorQualVars
SmartLabels
... Additional graphical parameters.

Details
The plotting procedure is similar to the one used for continuous biplots including the calibration of the environmental variables.

Value
No value returned

Author(s)
Jose Luis Vicente Villardon

References
CCA

See Also
plot.ContinuousBiplot

Examples
### Should be DIRECTLY executable !! ----
plot.ContinuousBiplot  Plots a biplot for continuous data.

Description

Plots a biplot for continuous data.

Usage

```r
## S3 method for class 'ContinuousBiplot'
plot(x, A1 = 1, A2 = 2, ShowAxis = FALSE, margin = 0,
     PlotVars = TRUE, PlotInd = TRUE, WhatInds = NULL,
     WhatVars = NULL, LabelVars = TRUE, LabelInd = TRUE,
     IndLabels = NULL, VarLabels = NULL, mode = "a",
     CexInd = NULL, CexVar = NULL, ColorInd = NULL, ColorVar = NULL,
     LabelPos = 1, SmartLabels = FALSE,
     MinQualityInds = 0, MinQualityVars = 0, dp = 0,
     PredPoints = 0, PlotAxis = FALSE, TypeScale = "Complete",
     ValuesScale = "Original", SizeQualInd = FALSE, SizeQualVars = FALSE,
     ColorQualInd = FALSE, ColorQualVars = FALSE,
     PchInd = NULL, PchVar = NULL, PlotClus = FALSE,
     TypeClus = "ch", ClustConf = 1,
     ClustCenters = FALSE, UseClusterColors = TRUE,
     PlotSupVars = FALSE, ShowBox = FALSE, nticks = 5,
     NonSelectedGray = FALSE, PlotUnitCircle = TRUE,
     PlotContribFA = TRUE, AddArrow = FALSE, ...)
```

Arguments

- `x`: An object of class "Biplot"
- `A1`: Dimension for the first axis. 1 is the default.
- `A2`: Dimension for the second axis. 2 is the default.
- `ShowAxis`: Logical variable to control if the coordinate axes should appear in the plot. The default value is FALSE because for most of the biplots its presence is irrelevant.
- `margin`: Margin for the labels in some of the biplot modes (percentage of the plot width). Default is 0. Increase the value if the labels are not completely plotted.
- `PlotVars`: Logical to control if the Variables (Columns) are plotted.
- `PlotInd`: Logical to control if the Individuals (Rows) are plotted.
- `WhatInds`: Logical vector to control what individuals (Rows) are plotted. (Can be also a binary vector)
- `WhatVars`: Logical vector to control what variables (Columns) are plotted. (Can be also a binary vector)
- `LabelVars`: Logical to control if the labels for the Variables are shown
- `LabelInd`: Logical to control if the labels for the individuals are shown
- `IndLabels`: A set of labels for the individuals. If NULL the default object labels are used
- `VarLabels`: A set of labels for the variables. If NULL the default object labels are used
- `mode`: Mode of the biplot: "p", "a", "b", "h", "ah" and "s".
- `CexInd`: Size for the symbols and labels of the individuals
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CexVar</td>
<td>Size for the symbols and labels of the variables</td>
</tr>
<tr>
<td>ColorInd</td>
<td>Color for the symbols and labels of the individuals</td>
</tr>
<tr>
<td>ColorVar</td>
<td>Color for the symbols and labels of the variables</td>
</tr>
<tr>
<td>LabelPos</td>
<td>Position of the labels in relation to the point. (See the graphical parameter pos)</td>
</tr>
<tr>
<td>SmartLabels</td>
<td>Plot the labels in a smart way</td>
</tr>
<tr>
<td>MinQualityInds</td>
<td>Minimum quality of representation for an individual to be plotted</td>
</tr>
<tr>
<td>MinQualityVars</td>
<td>Minimum quality of representation for a variable to be plotted</td>
</tr>
<tr>
<td>dp</td>
<td>A set of indices with the variables that will show the projections of the individuals</td>
</tr>
<tr>
<td>PredPoints</td>
<td>A vector with integers. The row points listed in the vector are projected onto all the variables.</td>
</tr>
<tr>
<td>PlotAxis</td>
<td>Not Used</td>
</tr>
<tr>
<td>TypeScale</td>
<td>Type of scale to use: &quot;Complete&quot;, &quot;StdDev&quot; or &quot;BoxPlot&quot;</td>
</tr>
<tr>
<td>ValuesScale</td>
<td>Values to show on the scale: &quot;Original&quot; or &quot;Transformed&quot;</td>
</tr>
<tr>
<td>SizeQualInd</td>
<td>Should the size of the row points be related to their qualities of representation (predictiveness)?</td>
</tr>
<tr>
<td>SizeQualVars</td>
<td>Should the size of the column points be related to their qualities of representation (predictiveness)?</td>
</tr>
<tr>
<td>ColorQualInd</td>
<td>Should the color of the row points be related to their qualities of representation (predictiveness)?</td>
</tr>
<tr>
<td>ColorQualVars</td>
<td>Should the color of the column points be related to their qualities of representation (predictiveness)?</td>
</tr>
<tr>
<td>PchInd</td>
<td>Symbol for the row points. See help(points) for details.</td>
</tr>
<tr>
<td>PchVar</td>
<td>Symbol for the column points. See help(points) for details.</td>
</tr>
<tr>
<td>PlotClus</td>
<td>Should the clusters be plotted?</td>
</tr>
<tr>
<td>TypeClus</td>
<td>Type of plot for the clusters. (&quot;ch&quot;- Convex Hull, &quot;el&quot;- Ellipse or &quot;st&quot;- Star)</td>
</tr>
<tr>
<td>ClustConf</td>
<td>Percent of points included in the cluster. only the ClusConf percent of the points nearest to the center will be used to calculate the cluster</td>
</tr>
<tr>
<td>ClustCenters</td>
<td>Should the cluster centers be plotted</td>
</tr>
<tr>
<td>UseClusterColors</td>
<td>Should the cluster colors be used in the plot</td>
</tr>
<tr>
<td>PlotSupVars</td>
<td>Should the supplementary variables be plotted?</td>
</tr>
<tr>
<td>ShowBox</td>
<td>Should a box around the points be plotted?</td>
</tr>
<tr>
<td>ticks</td>
<td>Number of ticks for the representation of the variables</td>
</tr>
<tr>
<td>NonSelectedGray</td>
<td>The nonselected individuals and variables are plotted in light gray colors</td>
</tr>
<tr>
<td>PlotUnitCircle</td>
<td>Plot the unit circle in the biplot for a Factor Analysis in which the length of the column arrows is smaller than 1 and is the quality of representation.</td>
</tr>
<tr>
<td>PlotContribFA</td>
<td>Plot circles in the biplot for a Factor Analysis with different values of the quality of representation.</td>
</tr>
<tr>
<td>AddArrow</td>
<td>Add an arrow to the representation of other modes of the biplot.</td>
</tr>
<tr>
<td>...</td>
<td>Any other graphical parameters</td>
</tr>
</tbody>
</table>
Details

Plots a biplot for continuous data. The Biplot for continuous data is taken as the basis of the plot. If there are a mixture of different types of variables (binary, nominal, abundance, ...) are added to the biplot as supplementary parts.

There are several modes for plotting the biplot. "p".- Points (Rows and Columns are represented by points)
"a".- Arrows (The traditional representation with points for rows and arrows for columns)
"b".- The arrows for the columns are extended to both extremes of the plot and labeled outside the plot area.
"h".- The arrows for the columns are extended to the positive extreme of the plot and labeled outside the plot area.
"ah".- Same as arrows but labeled outside the plot area.
"s".- The directions (or biplot axes) have a graded scale for prediction of the original values.

The `TypeScale` argument applies only to the "s" mode. There are three types:
"Complete".- An equally spaced scale covering the whole range of the data is calculates.
"StdDev".- Mean with one, two and three standard deviations
"BoxPlot".- Box-Plot like Scale (Median, 25 and 75 percentiles, maximum and minimum values.)

The `ValuesScale` argument applies only to the "s" mode and controls if the labels show the Original or Transformed values.

Some of the initial transformations are not compatible with some of the types of biplots and scales. For example, It is not possible to recover by projection the original values when you double centre de data. In that case you have the residuals for interaction and only the transformed values make sense.

It is possible to associate the color and the size of the points with the quality of representation. Bigger points correspond to better representation quality.

Value

No value Returned

Author(s)

Jose Luis Vicente Villardon

References

Examples

```r
data(Protein)
bip=PCA.Biplot(Protein[,3:11])
## Biplot with scales on the variables
plot(bip, mode="s", margin=0.2, ShowAxis=FALSE)
```

plot.ellipse  
*Plot a concentration ellipse.*

Description

Plot a concentration ellipse obtained from `concellipse`.

Usage

```r
## S3 method for class 'ellipse'
plot(x, add=TRUE, labeled=FALSE, center=FALSE, centerlabel="Center", initial=FALSE, ...)
```

Arguments

- `x`: An object with class `ellipse` obtained from `ConcEllipse`.
- `add`: Should the ellipse be added to the current plot?
- `labeled`: Should the ellipse be labelled with the confidence level?
- `center`: Should the center be plotted?
- `centerlabel`: Label for the center.
- `initial`: Should the initial data be plotted?
- `...`: Any other graphical parameter that can affects the plot (as color, etc...)

Details

Plots an ellipse containing a specified percentage of the data.

Value

No value returned

Author(s)

Jose Luis Vicente Villardon

References


Plots an External Logistic Biplot for binary data

**Description**

Plot of an External Binary Logistic Biplot with many arguments controlling different aspects of the representation.

**Usage**

```r
# S3 method for class 'External.Binary.Logistic.Biplot'
plot(x, F1 = 1, F2 = 2, ShowAxis = FALSE,
     margin = 0.2, WhatRows = NULL, WhatCols = NULL, RowLabels = NULL, ColLabels = NULL,
     RowColors = NULL, ColColors = NULL, Mode = "s", TickLength = 0.01, RowCex = 0.8,
     ColCex = 0.8, SmartLabels = FALSE, MinQualityRows = 0, MinQualityCols = 0, dp = 0,
     PredPoints = 0, SizeQualRows = FALSE, SizeQualCols = FALSE, ColorQualRows = FALSE,
     ColorQualCols = FALSE, PchRows = NULL, PchCols = NULL, PlotClus = FALSE,
     TypeClus = "ch", ClustConf = 1, Significant = FALSE, alpha = 0.05, Bonferroni = FALSE, ...)
```

**Arguments**

- `x` An object of type `External.Binary.Logistic.Biplot`
- `F1` Latent factor to represent at the X axis
- `F2` Latent factor to represent at the Y axis
- `ShowAxis` Should the axis be plotted?
- `margin` Margin for the labels in some of the biplot modes (percentage of the plot width). Default is 0. Increase the value if the labels are not completely plotted.
- `WhatRows` A binary vector (0 and 1) that indicates if each individual row should be plotted or not
- `WhatCols` A binary vector (0 and 1) that indicates if each individual column should be plotted or not
- `RowLabels` A vector of Labels for the rows if you do not want to use the data labels
- `ColLabels` A vector of Labels for the columns if you do not want to use the data labels
- `RowColors` A vector of colors for the rows
- `ColColors` A vector of colors for the rows
- `Mode` Mode of the biplot: "p", "a", "b", "ah" and "s". See details.

**See Also**

`ConcEllipse`
TickLength  Lenght of the tick marks. Depends on the scale of the graph.
RowCex  A scalar or a vector containing the sizes of the points and labels for the rows. Default value is 0.8 if the sizes are not provided.
ColCex  A scalar or a vector containing the sizes of the points and labels for the columns. Default value is 0.8 if the sizes are not provided.
SmartLabels  Plot the labels in a smart way
MinQualityRows  Minimum quality of representation for a row or individual to be plotted
MinQualityCols  Minimum quality of representation for a column or variable to be plotted
dp  "Drop Points" on the variables, a vector with integers. The row points are projected on the directions of the variables listed in the vector.
PredPoints  A vector with integers. The row points listed in the vector are projected onto all the variables.
SizeQualRows  Should the size of the row points be related to their qualities of representation (predictiveness)?
SizeQualCols  Should the size of the column points be related to their qualities of representation (predictiveness)?
ColorQualRows  Should the color of the row points be related to their qualities of representation (predictiveness)?
ColorQualCols  Should the color of the column points be related to their qualities of representation (predictiveness)?
PchRows  Symbol for the row points. See help(points) for details.
PchCols  Symbol for the column points. See help(points) for details.
PlotClus  Should the clusters be plotted?
TypeClus  Type of plot for the clusters. ("ch"- Convex Hull, "el"- Ellipse or "st"- Star)
ClustConf  Percent of points included in the cluster. only the ClusConf percent of the points nearest to the center will be used to calculate the cluster
Significant  If TRUE, only the significant variables are plotted
alpha  Significance Level
Bonferroni  Should the Bonferroni correction be used
...  Any other graphical parameter you want to use

**Details**

The logistic regression equation predicts the probability that a character will be present in an individual. Geometrically the y's can be represented as points in the reduced dimension space and the b's are the vectors showing the directions that best predict the probability of presence of each allele. For a complete explanation of the geometrical properties of the ELB see Vicente-Villardon et al (2006). The prediction of the probabilities is made in the same way as in a linear Biplot, i.e., the projection of a genotype point on the direction of an variable vector predicts the probability of presence of that variable in the individual. To facilitate the interpretation of the graph, fixed prediction probabilities points are situated on each allele vector. To simplify the graph, in our application, a vector joining the points for 0.5 and 0.75 are placed; this shows the cut point for prediction of presence and the direction of increasing probabilities. The length of the vector can be interpreted as an inverse measure of the discriminatory power of the alleles or bands, in the sense that shorter vectors correspond to alleles that better differentiate individuals. Two alleles pointing in the same direction are highly correlated, two alleles pointing in opposite directions are negatively correlated,
and two alleles forming an angle close to 90° are not correlated. A more complete scale with probabilities from 0.1 to 0.9 can also be plotted with this function. For each variable, the ordination diagram can be divided into two separate regions predicting presence or absence, the two regions are separated by the line that is perpendicular to the variable vector in the Biplot and cuts the vector in the point predicting 0.5. The variables associated to the configuration are those that predict the presences adequately. In a practical situation not all the variables are associated to the ordination. Due to the high number usually studied, it is convenient to situate on the graph only those that are related to the configuration, i.e., those that have an adequate goodness of fit after adjusting the logistic regression.

Value

No value returned

Author(s)

Jose Luis Vicente Villardon

References


See Also

ExternalBinaryLogisticBiplot

Examples

data(spiders)
dist=BinaryProximities(spiders)
pco=PrincipalCoordinates(dist)
pcobip=ExternalBinaryLogisticBiplot(pco)
pcobip=AddCluster2Biplot(pcobip, NGroups=3, ClusterType="hi")
op <- par(mfrow=c(1,2))
plot(pcobip, Mode="s", PlotClus = TRUE)
plot(pcobip$Dendrogram)
par(op)

plot.fraction

Plots a fraction of the data as a cluster

Description

Plots a convex hull or a star containing a specified percentage of the data. Used to plot clusters.
Usage

```r
## S3 method for class 'fraction'
plot(x, add = TRUE, center = FALSE, centerlabel = "Center", initial = FALSE, type = "ch", ...)
```

Arguments

- `x`: An object with class `fraction` obtained from `Fraction`.
- `add`: Should the fraction be added to the current plot?
- `center`: Should the center be plotted?
- `centerlabel`: Label for the center.
- `initial`: Should the initial data be plotted?
- `type`: Type of plot. Can be: "ch" - Convex Hull or "st" - Star (Joining each point with the center)
- `...`: Any other graphical parameter that can affects the plot (as color, etc ...)

Details

Plots a convex hull or a star containing a specified percentage of the data.

Value

No value returned

Author(s)

Jose Luis Vicente Villardon

See Also

`Fraction`

Examples

```r
a <- matrix(rnorm(50), 25, 2)
a2 <- Fraction(a, 0.7)
plot(a2, add=FALSE, type="ch", initial=TRUE, center=TRUE, col="blue")
plot(a2, add=TRUE, type="st", col="red")
```

Description

Plots an object of type MGC (Model-based Gaussian Clustering).

Usage

```r
## S3 method for class 'MGC'
plot(x, vars = NULL, groups = x$Classification, CexPoints = 0.2, Confidence = 0.95, ...)
```
Arguments

x  An object of type MGC
vars  A subset of indices of the variables to be plotted
groups  A factor containing groups to represent. Usually the clusters obtained from the algorithm.
CexPoints  Size of the points.
Confidence  Confidence of the ellipses
...  Anay additional graphical parameters

Details

Plots an object of type MGC (Model-based Gaussian Clustering) using a splom plot.

Value

No value returned

Author(s)

Jose Luis Vicente Villardon

Examples

data(iris)

plot.Ordinal.Logistic.Biplot

Plots an ordinal Logistic Biplot

Description

Plots an ordinal Logistic Biplot

Usage

## S3 method for class 'Ordinal.Logistic.Biplot'
plot(x, A1 = 1, A2 = 2, ShowAxis = FALSE, margin = 0, PlotVars = TRUE, PlotInd = TRUE, LabelVars = TRUE)

Arguments

x  Plots and object of type "Ordinal.Logistic.Biplot"
A1  First dimension to plot
A2  Second dimension to plot
ShowAxis  Should the axis be shown
margin  Margin for the graph (in order to have space for the variable levels)
PlotVars  Should the variables be plotted?
PlotInd  Should the individuals be plotted?
LabelVars  Should the variables be labelled?
### plot.Ordinal.Logistic.Biplot

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LabelInd</td>
<td>Should the variables be labelled?</td>
</tr>
<tr>
<td>mode</td>
<td>Mode of the biplot (see the classical biplot)</td>
</tr>
<tr>
<td>CexInd</td>
<td>Type of marker used for the individuals</td>
</tr>
<tr>
<td>CexVar</td>
<td>Type of marker used for the variables</td>
</tr>
<tr>
<td>ColorInd</td>
<td>Colors used for the individuals</td>
</tr>
<tr>
<td>ColorVar</td>
<td>Colors used for the variables</td>
</tr>
<tr>
<td>SmartLabels</td>
<td>Should smart placement for the labels be used?</td>
</tr>
<tr>
<td>MinQualityVars</td>
<td>Minimum quality of representation for a variable to be displayed</td>
</tr>
<tr>
<td>dp</td>
<td>Set of variables in which the individuals are projected</td>
</tr>
<tr>
<td>PredPoints</td>
<td>Set of points that will be projected on all the variables</td>
</tr>
<tr>
<td>PlotAxis</td>
<td>Should the axis be plotted?</td>
</tr>
<tr>
<td>TypeScale</td>
<td>See continuous biplots</td>
</tr>
<tr>
<td>ValuesScale</td>
<td>See continuous biplots</td>
</tr>
<tr>
<td>SizeQualInd</td>
<td>Should the size of the labels and points be related to the quality of representation for individuals?</td>
</tr>
<tr>
<td>SizeQualVars</td>
<td>Should the size of the labels and points be related to the quality of representation for variables?</td>
</tr>
<tr>
<td>ColorQualInd</td>
<td>Should the intensity of the color of the labels and points be related to the quality of representation for individuals?</td>
</tr>
<tr>
<td>ColorQualVars</td>
<td>Should the intensity of the color of the labels and points be related to the quality of representation for variables?</td>
</tr>
<tr>
<td>PchInd</td>
<td>Markers for the individuals</td>
</tr>
<tr>
<td>PchVar</td>
<td>Markers for the individuals</td>
</tr>
<tr>
<td>PlotClus</td>
<td>Should the added clusters for the individuals be plotted?</td>
</tr>
<tr>
<td>TypeClus</td>
<td>Type of plot for the clusters. The types are &quot;ch&quot;, &quot;el&quot; and &quot;st&quot; for &quot;Convex Hull&quot;, &quot;Ellipse&quot; and &quot;Star&quot; respectively.</td>
</tr>
<tr>
<td>ClustConf</td>
<td>Confidence level for the cluster</td>
</tr>
<tr>
<td>ClustCenters</td>
<td>Should the centers of the clusters be plotted</td>
</tr>
<tr>
<td>UseClusterColors</td>
<td>Should the colors of the clusters be used to plot the individuals.</td>
</tr>
<tr>
<td>ClustLegend</td>
<td>Should a legend for the clusters be added?</td>
</tr>
<tr>
<td>ClustLegendPos</td>
<td>Position of the legend</td>
</tr>
<tr>
<td>TextVarPos</td>
<td>Position of the labels for the variables</td>
</tr>
<tr>
<td>...</td>
<td>Any other additional parameters</td>
</tr>
</tbody>
</table>

#### Details

Plots an ordinal Logistic Biplot

#### Value

The plot ...

#### Author(s)

Jose Luis Vicente Villardon
References


See Also

plot.ContinuousBiplot

Examples

data(Doctors)
olb = OrdLogBipEM(Doctors, dim = 2, nnodes = 10, initial=4, tol = 0.001, maxiter = 100, penalization = 0.1, show=TRUE)
plot(olb, mode="s", ColorInd="gray", ColorVar=1:5)

plot.PCoABootstrap

Plots an object of class PCoABootstrap

Description

Plots an object of class PCoABootstrap

Usage

## S3 method for class 'PCoABootstrap'
plot(x, F1=1, F2=2, Move2Center=TRUE, BootstrapPlot="Ellipse", confidence=0.95, Colors=NULL, ...)

Arguments

- **x**: An object of class "PCoABootstrap"
- **F1**: First dimension to plot
- **F2**: Second dimension to plot
- **Move2Center**: Translate the ellipse center to the coordinates
- **BootstrapPlot**: Type of Bootstrap plot to draw: "Ellipse", "ConvexHull", "Star"
- **confidence**: Confidence level for the bootstrap plot
- **Colors**: Colors of the objects
- **...**: Additional parameters for graphical representations

Details

Draws the bootstrap confidence regions for the coordinates of the points obtained from a Principal Coordinates Analysis

Value

No value returned
Author(s)
Jose Luis Vicente Villardon

References

Examples
data(spiders)
Dis=BinaryProximities(spiders)
pco=PrincipalCoordinates(Dis, Bootstrap=TRUE, BootstrapType="Products")
plot(pco, Bootstrap=TRUE)

plot.Principal.Coordinates
Plots an object of class "Principal.Coordinates"

Description
Plots an object of class "Principal.Coordinates"

Usage
## S3 method for class 'Principal.Coordinates'
plot(x, F1 = 1, F2 = 2, LabelRows=TRUE, WhatRows = NULL, RowCex=1, RowPch=16,
    RowLabels = NULL, RowColors = NULL, SizeQualInd = FALSE, SmartLabels = FALSE,
    ColorQualInd = FALSE, ColorQual="black", PlotSup=TRUE, Bootstrap=FALSE,
    BootstrapPlot=c("Ellipse", "CovexHull", "Star"),
    PlotClus = FALSE, TypeClus = "ch", ClustConf = 1,
    ClustCenters = FALSE, UseClusterColors = TRUE, ...)

Arguments
x
Object of class "Principal.Coordinates"
F1
First dimension of the plot
F2
Second dimension of the plot
LabelRows
Controls if the points are labelled. Usually TRUE.
WhatRows
What Rows to plot. A vector of 0/1 elements. If NULL all rows are plotted
RowCex
Size of the points. Can be a single number or a vector.
RowPch
Symbols for the points.
RowLabels
Labels for the rows. If NULL row names of the data matrix are used.
RowColors
Colors for the rows. If NULL row default colors are assigned. Can be a single
value or a vector of colors.
SizeQualInd
Controls if the size of points depends on the quality of representation.
SmartLabels

Controls the way labels are plotted on the graph. If TRUE labels for points with positive x values are placed to the right of the point and labels for points with negative values to the left.

ColorQualInd

Controls if the color of the points depends on the quality of representation.

ColorQual

Darher color for the quality scale.

PlotSup

Controls if the supplementary points are plotted.

Bootstrap

Controls if the bootstrap points are plotted.

BootstrapPlot

Type of plot of the Bootstrap Information. The types are "Ellipse", "CovexHull" or "Star".

PlotClus

Should the clusters be plotted?

TypeClus

Type of plot for the clusters. ("ch"- Convex Hull, "el"- Ellipse or "st"- Star)

ClustConf

Percent of points included in the cluster. only the ClusConf percent of the points nearest to the center will be used to calculate the cluster

ClustCenters

Should the cluster centers be plotted

UseClusterColors

Should the cluster colors be used in the plot

... Additional parameters for graphical representations

Details

Graphical representation of an Principal coordinates Analysis controlling visual aspects of the plot as colors, symbols or sizes of the points.

Value

No value is returned

Author(s)

Jose Luis Vicente-Villardon

References


See Also

BinaryProximities

Examples

data(spiders)
dist=BinaryProximities(spiders)
pco=PrincipalCoordinates(dist)
plot(pco)
Description

Plots Simple Procrustes Analysis

Usage

```r
## S3 method for class 'Procrustes'
plot(x, F1=1, F2=2, ...)
```

Arguments

- `x`: Object of class "Procrustes"
- `F1`: First dimension of the plot
- `F2`: Second dimension of the plot
- `...`: Additional parameters for graphical representations

Details

Graphical representation of an Orthogonal Procrustes Analysis.

Value

No value is returned

Author(s)

Jose Luis Vicente-Villardon

See Also

`BinaryProximities`

Examples

```r
data(spiders)
dist=BinaryProximities(spiders)
pco=PrincipalCoordinates(dist)
plot(pco)
```
plot.Statis  

Plots a Statis Object

Description

Plots a Statis Object

Usage

```r
## S3 method for class 'Statis'
plot(x, A1 = 1, A2 = 2, ...)
```

Arguments

- `x`: A Statis object
- `A1`: First dimension of the plot
- `A2`: Second dimension of the plot
- `...`: Additional parameters

Details

Plots a Statis Object

Value

A biplot

Author(s)

Jose Luis Vicente Villardon

References


Examples

```r
data(Chemical)
x = Chemical[,5:16]
x=Convert2ThreeWay(x,Chemical$WEEKS, columns=FALSE)
stbip=StatisBiplot(x)
```
plot.Unfolding

Plots an Unfolding Representation

Description
Plots an Unfolding Representation

Usage
## S3 method for class 'Unfolding'
plot(x, A1 = 1, A2 = 2, ShowAxis = FALSE, margin = 0.1, PlotSites = TRUE, PlotSpecies = TRUE, PlotEnv = TRUE,
     SmartLabels = FALSE, PlotToL = FALSE, NNNI = TRUE)

Arguments
x
A1
A2
ShowAxis
margin
PlotSites
PlotSpecies
PlotEnv
LabelSites
LabelSpecies
LabelEnv
SpeciesQuality
MinQualityVars
dp
PlotAxis
TypeScale
ValuesScale
mode
CexSites
CexSpecies
CexVar
ColorSites
ColorSpecies
ColorVar
PchSites
PchSpecies
PchVar
SizeQualSites
plot3d.ContinuousBiplot

SizeQualSpecies
SizeQualVars
ColorQualSites
ColorQualSpecies

ColorQualVars
SmartLabels
PlotTol

Details
Plots an Unfolding Representation

plot3d.ContinuousBiplot

Plots a classical biplot for continuous data

Description
Plots a classical biplot for continuous data.

Usage

### S3 method for class 'ContinuousBiplot'

```r
plot3d(x, A1 = 1, A2 = 2, A3 = 3, ShowAxis = TRUE, margin = 0, PlotVars = TRUE, PlotInd = TRUE, WhatInds = NULL, WhatVars = NULL, LabelVars = TRUE, LabelInd = TRUE, ...)
```

Arguments

- **x**
  - An object of class "ContinuousBiplot"

- **A1**
  - Dimension for the first axis. 1 is the default.

- **A2**
  - Dimension for the second axis. 2 is the default.

- **A3**
  - Dimension for the third axis. 3 is the default.

- **ShowAxis**
  - Logical variable to control if the coordinate axes should appear in the plot. The default value is FALSE because for most of the biplots its presence is irrelevant.

- **margin**
  - Margin for the labels in some of the biplot modes (percentage of the plot width). Default is 0. Increase the value if the labels are not completely plotted.

- **PlotVars**
  - Logical to control if the Variables (Columns) are plotted.

- **PlotInd**
  - Logical to control if the Individuals (Rows) are plotted.

- **WhatInds**
  - Logical vector to control what individuals (Rows) are plotted. (Can be also a binary vector)

- **WhatVars**
  - Logical vector to control what variables (Columns) are plotted. (Can be also a binary vector)

- **LabelVars**
  - Logical to control if the labels for the Variables are shown

- **LabelInd**
  - Logical to control if the labels for the individuals are shown
IndLabels: A set of labels for the individuals. If NULL the default object labels are used.
VarLabels: A set of labels for the variables. If NULL the default object labels are used.
mode: Mode of the biplot: "p", "a", "b", "h", "ah" and "s".
CexInd: Size for the symbols and labels of the individuals.
CexVar: Size for the symbols and labels of the variables.
ColorInd: Color for the symbols and labels of the individuals.
ColorVar: Color for the symbols and labels of the variables.
LabelPos: Position of the labels in relation to the point. (See the graphical parameter pos.)
SmartLabels: Plot the labels in a smart way.
MinQualityInds: Minimum quality of representation for an individual to be plotted.
MinQualityVars: Minimum quality of representation for a variable to be plotted.
dp: A set of indices with the variables that will show the projections of the individuals.
PredPoints: A vector with integers. The row points listed in the vector are projected onto all the variables.
PlotAxis: Not Used.
TypeScale: Type of scale to use: "Complete", "StdDev" or "BoxPlot".
ValuesScale: Values to show on the scale: "Original" or "Transformed".
SizeQualInd: Should the size of the row points be related to their qualities of representation (predictiveness)?
SizeQualVars: Should the size of the column points be related to their qualities of representation (predictiveness)?
ColorQualInd: Should the color of the row points be related to their qualities of representation (predictiveness)?
ColorQualVars: Should the color of the column points be related to their qualities of representation (predictiveness)?
PchInd: Symbol for the row points. See help(points) for details.
PchVar: Symbol for the column points. See help(points) for details.
PlotClus: Should the clusters be plotted?
TypeClus: Type of plot for the clusters. ("ch"- Convex Hull, "el"- Ellipse or "st"- Star)
ClustConf: Percent of points included in the cluster. only the ClusConf percent of the points nearest to the center will be used to calculate the cluster.
ClustCenters: Should the cluster centers be plotted.
UseClusterColors: Should the cluster colors be used in the plot.
PlotSupVars: Should the supplementary variables be plotted?
...: Any other graphical parameters.

Details:
The parameters are the same as the ones for the 2D biplot.

Value:
A 3D Biplot.
plot3dCanonicalBiplot

Author(s)
Jose Luis Vicente Villardon

See Also
plot.ContinuousBiplot

Examples

data(Protein)
bip=PCA.Biplot(Protein[,3:11])
## Biplot with scales on the variables
plot3d.ContinuousBiplot(bip, mode="s", margin=0.2, ShowAxis=FALSE)

plot3dCanonicalBiplot  3D Canonical Biplot

Description
Plots a 3D Canonical Biplot

Usage
plot3dCanonicalBiplot(Bip, A1 = 1, A2 = 2, A3 = 3, ScaleGraph = TRUE, PlotGroups = TRUE, PlotVars = TRUE, PlotInd = TRUE, LabelInd = TRUE, CexGroup = NULL, PchGroup = NULL, margin = NULL, AddLegend = TRUE, ShowAxes = TRUE, LabelAxes = TRUE, LabelGroups = TRUE, PlotCircle = FALSE, ConvexHulls = FALSE)

Arguments
Bip  An object of class "Canonical Biplot"
A1   Dimension for the first axis. 1 is the default.
A2   Dimension for the second axis. 2 is the default.
A3   Dimension for the third axis. 3 is the default.
ScaleGraph   Reescale the coordinates to optimal matching.
PlotGroups   Shoud the group centers be plotted?
PlotVars   Should the variables be plotted?
PlotInd   Should the individuals be plotted?
LabelInd   Should the individuals be labeled?
CexGroup   Sizes of the points for the groups
PchGroup   Markers for the group
margin   margin for the graph
AddLegend   Should a legend with the groups be added?
ShowAxes   Should outside axes be shown?
LabelAxes   Should outside axes be labelled?
LabelGroups   Should the groups be labeled?
PlotCircle   Should the confidence regions for the groups be plotted?
ConvexHulls   Should the convex hulls containing the individuals for each group be plotted?
plot3dCanonicalBiplot

Type Circle

Type of confidence region: Univariate (U), Bonferroni (B), Multivariate (M) or Classical (C)

Color Groups

User colors for the groups. Default colors will be used if NULL.

Color Vars

User colors for the variables. Default colors will be used if NULL.

Legend Pos

Position of the legend.

Color Ind

User colors for the individuals. Default colors will be used if NULL.

Voronoi

Should the voronoi diagram with the prediction regions for each group be plotted?

Mode

Mode of the biplot: "p", "a", "b", "h", "ah" and "s".

Type Scale

Type of scale to use: "Complete", "StdDev" or "BoxPlot"

Values Scale

Values to show on the scale: "Original" or "Transformed"

Min Quality Vars

Minimum quality of representation for a variable to be plotted

dpg

A set of indices with the variables that will show the projections of the groups

dpi

A set of indices with the variables that will show the projections of the individuals

Pred Points

A vector with integers. The group centers listed in the vector are projected onto all the variables.

Plot Axis

Not Used

Cex Ind

Size of the points for individuals.

Cex Var

Size of the points for variables.

Pch Ind

Markers of the points for individuals.

Pch Var

Markers of the points for variables.

Color Var

Colors of the points for variables.

Show Axis

Should axis scales be shown?

Voronoicolor

Color for the Voronoi diagram

... Any additional graphical parameters

Details

The parameters are the same as in the 2D Canonical Biplot.

Value

A 3D Canonical Biplot

Author(s)

Jose Luis Vicente Villardon

See Also

plot Canonical Biplot

Examples

data(wine)
X=wine[,4:21]
canbip=CanonicalBiplot(X, group=wine$Group)
plot3dCanonicalBiplot(canbip, Type Circle="M")
PlotBiplotClusters  

*Plot clusters on a biplot.*

**Description**

Highlights several groups or clusters on a biplot representation.

**Usage**

```r
PlotBiplotClusters(A, Groups = ones(c(nrow(A), 1)), TypeClus = "st",
                   ClusterColors = NULL, ClusterNames = NULL, centers =
                   TRUE, ClustConf = 1, Legend = TRUE, LegendPos =
                   "topleft", ...)
```

**Arguments**

- `A` Coordinates of the points in the scattergram
- `Groups` Factor defining the groups to be highlighted
- `TypeClus` Type of representation of the clusters. For the moment just a convex hull but in the future ellipses and stars will be added.
- `ClusterColors` A vector of colors with as many elements as clusters. If NULL the function selects the rainbow colors.
- `ClusterNames` A vector of names with as many elements as clusters.
- `centers` Logical variable to control if centers of the clusters are plotted
- `ClustConf` Percent of points included in the cluster. Only the ClusConf percent of the points nearest to the center will be used to calculate the cluster
- `Legend` Should a legend be plotted
- `LegendPos` Position of the legend.
- `...` Any other graphical parameters

**Details**

The clusters to plot should be added to the biplot object using the function `AddCluster2Biplot`.

**Value**

It takes effects on a plot

**Author(s)**

Jose Luis Vicente Villardon

**See Also**

- `AddCluster2Biplot`
**Examples**

```r
data(iris)
bipPCA.Biplot(iris[,1:4])
bip=AddCluster2Biplot(bip, NGroups=3, ClusterType="us", Groups=iris[,5], Original=FALSE)
plot(bip, PlotClus = TRUE)
```

---

**Description**

Plot the response functions along the directions of best fit.

**Usage**

```r
PlotOrdinalResponses(olb, A1 = 1, A2 = 2, inf = -12, sup = 12, Legend = TRUE, WhatVars=NULL)
```

**Arguments**

- `olb`: An object of class "Ordinal.Logistic.Biplot"
- `A1`: First dimension of the plot.
- `A2`: Second dimension of the plot.
- `inf`: Lower limit of the representation.
- `sup`: Upper limit of the representation.
- `Legend`: Should a legend be plotted.
- `WhatVars`: A vector with the numbers of the variables to be plotted. If NULL all the variables are plotted.

**Details**

Plot the response functions along the directions of best fit for the selected dimensions.

**Value**

A plot describing the behaviour of the variable.

**Author(s)**

Jose Luis Vicente Villardon

**Examples**

```r
data(Doctors)
olb = OrdLogBipEM(Doctors, dim = 2, nnodes = 10, initial=4, tol = 0.001, maxiter = 100, penalization = 0.1, show=TRUE)
PlotOrdinalResponses(olb, WhatVars=c(1,2,3,4))
```
Partial Least Squares Regression (PLSR)

Description

Fits a Partial Least Squares Regression (PLSR) to a set of two continuous data matrices.

Usage

```r
PLSRfit(Y, X, s = 2, center = TRUE, scale = TRUE, tolerance = 5e-06, maxiter = 100, show = FALSE)
```

Arguments

- `Y`: The matrix of dependent variables
- `X`: The Matrix of Independent Variables
- `s`: Dimension of the solution. The default is 2
- `center`: Logical. Should the original data be centered.
- `scale`: Logical. Should the original data be scaled.
- `tolerance`: Tolerance for the algorithm.
- `maxiter`: Maximum number of iterations for the algorithm.
- `show`: Logical. Should the calculation process be shown on the screen.

Details

Fits a Partial Least Squares Regression (PLSR) to a set of two continuous data matrices.

Value

An object of class "PLSR"

- `Method`: PLSR1
- `X`: Independent Variables
- `Y`: Dependent Variables
- `center`: Are data centered?
- `scale`: Are data scaled?
- `ScaledX`: Scaled Independent Variables
- `ScaledY`: Scaled Dependent Variables
- `XScores`: Scores for the Independent Variables
- `XWeights`: Weights for the Independent Variables - coefficients of the linear combination
- `XLoadings`: Factor loadings for the Independent Variables
- `YScores`: Scores for the Dependent Variables
- `YWeights`: Weights for the Dependent Variables - coefficients of the linear combination
- `YLoadings`: Factor loadings for the Dependent Variables
- `XStructure`: Structure Correlations for the Independent Variables
- `YStructure`: Structure Correlations for the Dependent Variables
- `XXStructure`: Structure Correlations two groups
**Political Figures in the USA**

**Description**

Does the American public actively differentiate political stimuli along ideological lines?. Dissimilarities among 13 political figures in the USA.

**Usage**

```r
data("PoliticalFigures")
```

**Format**

A data frame with the dissimilarities among 13 political figures in the USA.

- `G._W._Bush`: a numeric vector with the dissimilarities with the other figures
- `John_Kerry`: a numeric vector with the dissimilarities with the other figures
- `Ralph_Nader`: a numeric vector with the dissimilarities with the other figures
- `Dick_Cheney`: a numeric vector with the dissimilarities with the other figures
- `John_Edwards`: a numeric vector with the dissimilarities with the other figures
- `Laura_Bush`: a numeric vector with the dissimilarities with the other figures
- `Hillary_Clinton`: a numeric vector with the dissimilarities with the other figures
- `Bill_Clinton`: a numeric vector with the dissimilarities with the other figures
- `Colin_Powell`: a numeric vector with the dissimilarities with the other figures
- `John_Ashcroft`: a numeric vector with the dissimilarities with the other figures
- `John_McCain`: a numeric vector with the dissimilarities with the other figures
- `Democ._Party`: a numeric vector with the dissimilarities with the other figures
- `Repub._Party`: a numeric vector with the dissimilarities with the other figures

**Details**

We have taken information from the 2004 CPS American National Election Study. Specifically 711 NES respondents’ feeling thermometer ratings of thirteen prominent political figures from the period of the 2004 election: George W. Bush; John Kerry; Ralph Nader; Richard Cheney; John Edwards; Laura Bush; Hillary Clinton; Bill Clinton; Colin Powell; John Ashcroft; John McCain; the Democratic party; and the Republican party. With the respondent scores, a dissimilarity among each pair of figures.
PrettyTicks

Calculates loose axis ticks and labels using nice numbers

Description

Calculates axis ticks and labels using nice numbers

Usage

PrettyTicks(min = -3, max = 3, ntick = 5)

Arguments

- min: Minimum value on the axis
- max: Maximum value on the axis
- ntick: Approximated number of desired ticks

Details

 Calculates axis ticks and labels using nice numbers. The resulting labels are known as loose labels.

Value

A list with the following fields

- ticks: Ticks for the axis
- labels: The corresponding labels

Author(s)

Jose Luis Vicente Villardon

References

Principal Coordinates

See Also

NiceNumber

Examples

prettyTicks(-4, 4, 5)

PrincipalCoordinates principal Coordinates Analysis

Description

Principal coordinates Analysis for a matrix of proximities obtained from binary, categorical, continuous or mixed data

Usage

PrincipalCoordinates(Proximities, dimension = 2, tolerance = 1e-04, Bootstrap=FALSE, BootstrapType=c("HMTL", "TL", "UI"), bootstrap=false, bootstrapType=c("HMTL", "TL", "UI"), nb, procrustesRot=TRUE, bootstrapMethod=c("HMTL", "TL", "UI"))

Arguments

Proximities An object of class proximities.
dimension Dimension of the solution
tolerance Tolerance for the eigenvalues
Bootstrap Should Bootstrap be calculated?
BootstrapType Bootstrap on the residuals of the "distance" or "scalar products" matrix.
nb Number of Bootstrap replications
ProcrustesRot Should each replication be rotated to match the initial solution?
bootstrapMethod Bootstrap on the residuals of the "distance" or "scalar products" matrix.

Details

Principal Coordinates Analysis for a proximity matrix previously calculated from a matrix of raw data or directly observed proximities.

Value

An object of class PrincipalCoordinates. The function adds the information of the Principal Coordinates to the object of class proximities. Together with the information about the proximities the object has:

Analysis The type of analysis performed, "Principal Coordinates" in this case
Eigenvalues The eigenvalues of the PCoA
Inertia The Inertia of the PCoA
RowCoordinates Coordinates for the objects in the PCoA
RowQualities Qualities of representation for the objects in the PCoA
RawStress Raw Stress values
print.MGC

stress1  stress formula 1
stress2  stress formula 2
ssstress1  sstress formula 1
ssstress2  sstress formula 2
rsq  Squared correlation between disparities and distances
Spearman  Spearman correlation between disparities and distances
Kendall  Kendall correlation between disparities and distances
BootstrapInfo  The result of the bootstrap calculations

Author(s)

Jose Luis Vicente-Villardon

References


See Also

BinaryProximities, BootstrapDistance, BootstrapDistance, BinaryProximities

Examples

data(spiders)
Dis=BinaryProximities(spiders)
pco=PrincipalCoordinates(Dis)
Dis=BinaryProximities(spiders)
pco=PrincipalCoordinates(Dis, Bootstrap=TRUE)

print.MGC  Prints the results of Model-Based Gaussian Clustering algorithms

Description

Prints the results of Model-Based Gaussian Clustering algorithms

Usage

## S3 method for class 'MGC'
print(x, ...)

Arguments

x          An object of class "MGC"
...        Any additional parameters

Details

Prints the results of Model-Based Gaussian Clustering algorithms

Value

No value returned

Author(s)

Jose Luis Vicente Villardon

Examples

```r
### Should be DIRECTLY executable !! ----
### --> Define data, use random,
### or do help(data=index) for the standard data sets.
```

---

### Protein

#### Protein consumption data.

Description

Protein consumption in twenty-five European countries for nine food groups.

Usage

data(Protein)

Format

A data frame with 25 observations on the following 11 variables.

- **Comunist**: a factor with levels No Yes
- **Region**: a factor with levels North Center South
- **Red_Meat**: a numeric vector
- **White_Meat**: a numeric vector
- **Eggs**: a numeric vector
- **Milk**: a numeric vector
- **Fish**: a numeric vector
- **Cereal**: a numeric vector
- **Starch**: a numeric vector
- **Nuts**: a numeric vector
- **Fruits_Vegetables**: a numeric vector
Details

These data measure protein consumption in twenty-five European countries for nine food groups. It is possible to use multivariate methods to determine whether there are groupings of countries and whether meat consumption is related to that of other foods.

Source

http://lib.stat.cmu.edu/DASL/Datafiles/Protein.html

References


Examples

data(Protein)
## maybe str(Protein) ; plot(Protein) ...

---

**Sugar Cane Data**

**Description**

Molecular characteristics of 50 varieties of sugar cane.

**Usage**

data(RAPD)

**Format**

A data frame with 50 observations on 168 variables. 1-120: Random amplified polymorphic DNA and 121-168: Microsatellites

**Details**

Data are codified as presence or absence of the dominant marker.

**Examples**

data(RAPD)
## maybe str(RAPD) ; plot(RAPD) ...
RemoveRowsWithNaNs

Remove rows that contains NaNs (missing data)

Description
Remove rows that contains NaNs to obtain a matrix without missing data

Usage
RemoveRowsWithNaNs(x, cols = NULL)

Arguments
- x The matrix to be arranged
- cols A set of columns to check as a vector of integers

Details
Remove rows that contains NaNs to obtain a matrix without missing data

Value
x Matrix without missing data

Author(s)
Jose Luis Vicente-Villardon

riano
Ecological data from Riano (Spain)

Description
Ecological data from Riano (Spain)

Usage
data("riano")

Format
A data frame with 70 observations on the following 25 variables.
- Week a factor with levels A B C D E F G H I J
- Depth a factor with levels 0 2 5 10 15 20 Bottom
- Cianof a numeric vector
- Crisof a numeric vector
- Haptof a numeric vector
- Crasp a numeric vector
Cripto a numeric vector
Dinof a numeric vector
Diatom a numeric vector
Euglen a numeric vector
Prasin a numeric vector
Clorof a numeric vector
Zigofi a numeric vector
Xantof a numeric vector
Malgas a numeric vector
Ta a numeric vector
X02 a numeric vector
pH a numeric vector
COND a numeric vector
SiO2 a numeric vector
P.PO4 a numeric vector
Chla a numeric vector
Chlb a numeric vector
Chlc a numeric vector
IM a numeric vector

Details
Ecological data from Riano (Spain). Abundance of several algae taxonomic groups and several environmental variables

Source
Department of Ecology. University of Leon. Spain

Examples

data(riano)
## maybe str(riano); plot(riano) ...

RidgeBinaryLogistic Ridge Binary Logistic Regression for Binary data

Description
This function performs a logistic regression between a dependent binary variable y and some independent variables x, solving the separation problem in this type of regression using ridge penalization.

Usage
RidgeBinaryLogistic(y, xd, freq = NULL, tolerance = 1e-05, maxiter = 100, penalization = 0.2, cte=0)
Ridge Binary Logistic

Arguments

- \( y \)  A binary dependent variable
- \( x_d \)  A set of independent variables
- \( freq \)  frequencies for each observation (usually 1)
- \( tolerance \)  Tolerance for convergence
- \( maxiter \)  Maximum number of iterations
- \( penalization \)  Ridge penalization: a non negative constant. Penalization used in the diagonal matrix to avoid singularities.
- \( cte \)  Should the model have a constant?

Details

Logistic Regression is a widely used technique in applied work when a binary, nominal or ordinal response variable is available, due to the fact that classical regression methods are not applicable to this kind of variables. The method is available in most of the statistical packages, commercial or free. Maximum Likelihood together with a numerical method as Newton-Raphson, is used to estimate the parameters of the model. In logistic regression, when in the space generated by the independent variables there are hyperplanes that separate among the individuals belonging to the different groups defined by the response, maximum likelihood does not converge and the estimations tend to the infinity. That is known in the literature as the separation problem in logistic regression. Even when the separation is not complete, the numerical solution of the maximum likelihood has stability problems. From a practical point of view, that means the estimated model is not accurate precisely when there should be a perfect, or almost perfect, fit to the data.

The problem of the existence of the estimators in logistic regression can be seen in Albert (1984), a solution for the binary case, based on the Firth method, Firth (1993) is proposed by Heinze(2002). The extension to nominal logistic model was made by Bull (2002). All the procedures were initially developed to remove the bias but work well to avoid the problem of separation. Here we have chosen a simpler solution based on ridge estimators for logistic regression Cessie(1992).

Rather than maximizing \( L_j(G | b_{j0}, B_j) \) we maximize

\[
L_j(G | b_{j0}, B_j) - \lambda (\|b_{j0}\| + \|B_j\|)
\]

Changing the values of \( \lambda \) we obtain slightly different solutions not affected by the separation problem.

Value

An object of class \texttt{RidgeBinaryLogistic} with the following components

- \( \text{beta} \)  Estimates of the coefficients
- \( \text{fitted} \)  Fitted probabilities
- \( \text{residuals} \)  Residuals of the model
- \( \text{Prediction} \)  Predictions of presences and absences
- \( \text{Covariances} \)  Covariances among the estimates
- \( \text{Deviance} \)  Deviance of the current model
- \( \text{NullDeviance} \)  Deviance of the null model
- \( \text{Dif} \)  Difference between the deviances of the current and null models
df  Degrees of freedom of the difference
p   p-value
CoxSnell  Cox-Snell pseudo R-squared
Nagelkerke  Nagelkerke pseudo R-squared
MacFaden  MacFaden pseudo R-squared
R2  Pseudo R-squared using the residuals
Classification  Classification table
PercentCorrect  Percentage of correct classification

Author(s)

Jose Luis Vicente Villardon

References

RidgeBinaryLogisticFit


Examples

```r
data(spiders)
dist=BinaryProximities(spiders)
pco=PrincipalCoordinates(dist)
x=pco$RowCoordinates
y=as.numeric(spiders[,2])-1
fit=RidgeBinaryLogistic(y,x)
fit
```

RidgeBinaryLogisticFit

*Fits a binary logistic regression with ridge penalization*

Description

This function fits a logistic regression between a dependent variable y and some independent variables x, and solves the separation problem in this type of regression using ridge regression and penalization.

Usage

```r
RidgeBinaryLogisticFit(y, xd, freq, tolerance = 1e-05, maxiter = 100, penalization = 0.2)
```
RidgeMultinomialLogisticFit

Arguments

- **y**: A vector with the values of the dependent variable.
- **xd**: A matrix with the independent variables.
- **freq**: Frequencies of each pattern.
- **tolerance**: Tolerance for the iterations.
- **maxiter**: Maximum number of iterations for convergence.
- **penalization**: Penalization used in the diagonal matrix to avoid singularities.

Details

Fits a binary logistic regression with ridge penalization.

Value

The parameters of the fit.

Author(s)

Jose Luis Vicente Villardon

See Also

RidgeBinaryLogistic

Examples

```R
# Should be DIRECTLY executable ! ! ----
```

```R
RidgeMultinomialLogisticFit

Multinomial logistic regression with ridge penalization
```

Description

This function does a logistic regression between a dependent variable y and some independent variables x, and solves the separation problem in this type of regression using ridge regression and penalization.

Usage

```R
RidgeMultinomialLogisticFit(y, x, penalization = 0.2, tol = 1e-04, maxiter = 200, show = FALSE)
```

Arguments

- **y**: Dependent variable.
- **x**: A matrix with the independent variables.
- **penalization**: Penalization used in the diagonal matrix to avoid singularities.
- **tol**: Tolerance for the iterations.
- **maxiter**: Maximum number of iterations.
- **show**: Should the iteration history be printed?
Details

The problem of the existence of the estimators in logistic regression can be seen in Albert (1984), a solution for the binary case, based on the Firth’s method, Firth (1993) is proposed by Heinze (2002). The extension to nominal logistic model was made by Bull (2002). All the procedures were initially developed to remove the bias but work well to avoid the problem of separation. Here we have chosen a simpler solution based on ridge estimators for logistic regression Cessie (1992).

Rather than maximizing \( L_j(\mathbf{G} | \mathbf{b}_j, \mathbf{B}_j) \) we maximize

\[
L_j(\mathbf{G} | \mathbf{b}_j, \mathbf{B}_j) - \lambda (\|\mathbf{b}_j\| + \|\mathbf{B}_j\|)
\]

Changing the values of \( \lambda \) we obtain slightly different solutions not affected by the separation problem.

Value

An object of class "rm1r" with components

- fitted: Matrix with the fitted probabilities
- cov: Covariance matrix among the estimates
- \( \mathbf{Y} \): Indicator matrix for the dependent variable
- beta: Estimated coefficients for the multinomial logistic regression
- stderr: Standard error of the estimates
- logLik: Logarithm of the likelihood
- Deviance: Deviance of the model
- AIC: Akaike information criterion indicator
- BIC: Bayesian information criterion indicator

Author(s)

Jose Luis Vicente-Villardon

References


Firth, D.(1993), Bias reduction of maximum likelihood estimates, Biometrika 80(1), 27–38


Examples

# No examples yet
Ridge Multinomial Logistic Regression

Description
Function that calculates an object with the fitted multinomial logistic regression for a nominal variable. It compares with the null model, so that we will be able to compare which model fits better the variable.

Usage
RidgeMultinomialLogisticRegression(formula, data, penalization = 0.2, cte = TRUE, tol = 1e-04, maxiter = 200, showIter = FALSE)

Arguments
- formula: The usual formula notation (or the dependent variable)
- data: The dataframe used by the formula. (or a matrix with the independent variables).
- penalization: Penalization used in the diagonal matrix to avoid singularities.
- cte: Should the model have a constant?
- tol: Value to stop the process of iterations.
- maxiter: Maximum number of iterations.
- showIter: Should the iteration history be printed?.

Value
An object that has the following components:

- fitted: Matrix with the fitted probabilities
- cov: Covariance matrix among the estimates
- Y: Indicator matrix for the dependent variable
- beta: Estimated coefficients for the multinomial logistic regression
- stderr: Standard error of the estimates
- logLik: Logarithm of the likelihood
- Deviance: Deviance of the model
- AIC: Akaike information criterion indicator
- BIC: Bayesian information criterion indicator
- NullDeviance: Deviance of the null model
- Difference: Difference between the two deviance values
- df: Degrees of freedom
- p: p-value associated to the chi-squared estimate
- CoxSnell: Cox and Snell pseudo R squared
- Nagelkerke: Nagelkerke pseudo R squared
- MacFaden: MacFaden pseudo R squared
- Table: Cross classification of observed and predicted responses
- PercentCorrect: Percentage of correct classifications
RidgeOrdinalLogistic

Author(s)
Jose Luis Vicente-Villardon

References

See Also
RidgeMultinomialLogisticFit

Examples

data(Protein)
y=Protein[,2]
x=Protein[,c(3,11)]
rmlr = RidgeMultinomialLogisticRegression(y,x,penalization=0.0)
summary(rmlr)

RidgeOrdinalLogistic  Ordinal logistic regression with ridge penalization

Description
This function performs a logistic regression between a dependent ordinal variable y and some independent variables x, and solves the separation problem using ridge penalization.

Usage
RidgeOrdinalLogistic(y, x, penalization = 0.1, tol = 1e-04, maxiter = 200, show = FALSE)

Arguments
y  Dependent variable.
x  A matrix with the independent variables.
penalization  Penalization used to avoid singularities.
tol  Tolerance for the iterations.
maxiter  Maximum number of iterations.
show  Should the iteration history be printed?.
Details

The problem of the existence of the estimators in logistic regression can be seen in Albert (1984); a solution for the binary case, based on the Firth’s method, Firth (1993) is proposed by Heinze(2002). All the procedures were initially developed to remove the bias but work well to avoid the problem of separation. Here we have chosen a simpler solution based on ridge estimators for logistic regression Cessie(1992).

Rather than maximizing \( L_j(G | b_{j0}, B_j) \) we maximize

\[
L_j(G | b_{j0}, B_j) - \lambda (\|b_{j0}\| + \|B_j\|)
\]

Changing the values of \( \lambda \) we obtain slightly different solutions not affected by the separation problem.

Value

An object of class "pordlogist". This has components:

- nobs: Number of observations
- J: Maximum value of the dependent variable
- nvar: Number of independent variables
- fitted.values: Matrix with the fitted probabilities
- pred: Predicted values for each item
- Covariances: Covariances matrix
- clasif: Matrix of classification of the items
- PercentClassif: Percent of good classifications
- coefficients: Estimated coefficients for the ordinal logistic regression
- thresholds: Thresholds of the estimated model
- logLik: Logarithm of the likelihood
- penalization: Penalization used to avoid singularities
- Deviance: Deviance of the model
- DevianceNull: Deviance of the null model
- Dif: Difference between the two deviances values calculated
- df: Degrees of freedom
- pval: p-value of the contrast
- CoxSnell: Cox-Snell pseudo R squared
- Nagelkerke: Nagelkerke pseudo R squared
- MacFaden: Nagelkerke pseudo R squared
- iter: Number of iterations made

Author(s)

Jose Luis Vicente-Villardon
scores.CCA.sol

References


Examples

data(Doctors)
olb = OrdLogBipEM(Doctors, dim = 2, nnodos = 10,
                    tol = 0.001, maxiter = 100, penalization = 0.2)
model = RidgeOrdinalLogistic(Doctors[, 1], olb$RowCoordinates, tol = 0.001,
                             maxiter = 100, penalization = 0.2)
model

scores.CCA.sol

Extract the scores of a CCA solution object

Description

Extract the scores of a CCA solution object

Usage

clccs.CCA.sol(CCA.sol)

Arguments

CCA.sol

Details

Extract the scores of a CCA solution object

Value

The species, sites and environmental variables scores of a CCA solution

Author(s)

Jose Luis Vicente Villardon

See Also

CCA
Examples

#### Should be DIRECTLY executable !! ----

SeparateVarTypes

**Separation of different types of variables into a list**

**Description**

The procedure creates a list in which each field contains the variables of the same type.

**Usage**

SeparateVarTypes(X, TypeVar = NULL, TypeFit = NULL)

**Arguments**

- **X** A data frame
- **TypeVar** A vector of characters defining the type of each variable. If not provided the procedure tries to guess the type of each variable. See details for types
- **TypeFit** A vector of characters defining the type of fit for each variable. If not provided the procedure tries to guess the type of fit for each variable. See details for types

**Details**

The procedure creates a list in which each field contains the variables of the same type. The type of Variable can be specified in a vector TypeVar and the type of fit in a vector TypeFit. The TypeVar is a vector of characters with as many components as variables with types coded as:

- "c" - Continuous (1)
- "b" - Binary (2)
- "n" - Nominal (3)
- "o" - Ordinal (4)
- "f" - Frequency (5)
- "a" - Abundance (5)

Numbers rather than characters can also be used. Unless specified in TypeVar, numerical variables are "Continuous", factors are "Nominal", ordered factors are "Ordinal". Factors with just two values are considered as "Binary", "Frequencies" and "abundances" should be specified by the user. If Typevar has length 1, all the variables are supposed to have the same type.

The typeFit is a vector of characters containing the type of fit used for each variab, coded as:

- "a" - Average (1)
- "wa" - Weighted Average (2)
- "r" - Regression (Linear or logistic depending on the type of variable) (3)
- "g" - Gaussian (Equal tolerances) (4)
- "g1" - Gaussian (Different tolerances) (5)

Numbers rather than characters can also be used. Unless specified numerical variables are fitted with linear regression, factors with logistic biplots, frequencies with weighted averages and abundances with gaussian regression.
Value

A list containing the following fields

- **Continuous**: A list containing a data frame with the numeric variables and a character vector with the type of fit for each variable
- **Binary**: A list containing a data frame with the binary variables and a character vector with the type of fit for each variable
- **Nominal**: A list containing a data frame with the nominal variables and a character vector with the type of fit for each variable
- **Ordinal**: A list containing a data frame with the ordinal variables and a character vector with the type of fit for each variable
- **Frequency**: A list containing a data frame with the frequency variables and a character vector with the type of fit for each variable
- **Abundance**: A list containing a data frame with the abundance variables and a character vector with the type of fit for each variable

**Author(s)**

Jose Luis Vicente Villardon

**Examples**

```r
data(Protein)
SepData=SeparateVarTypes(Protein)
SepData
```

**Description**

Simple Procrustes Analysis for two matrices

**Usage**

```r
SimpleProcrustes(X, Y, centre = FALSE)
```

**Arguments**

- **X**: Matrix of the first configuration.
- **Y**: Matrix of the second configuration.
- **centre**: Should the matrices be centred before the calculations?

**Details**

Orthogonal Procrustes Analysis for two configurations $X$ and $Y$. The first configuration $X$ is used as a reference and the second, $Y$, is transformed to match the reference as much as possible. $X = sY^T + 1t + E = Z + E$
Value

An object of class Procrustes. This has components:

- X: First Configuration
- Y: Second Configuration
- Yrot: Second Configuration after the transformation
- T: Rotation Matrix
- t: Translation Vector
- s: Scale Factor
- rss: Residual Sum of Squares
- fit: Goodness of fit as percent of explained variance
- correlations: Correlations among the columns of X and Z

Author(s)

Jose Luis Vicente-Villardon

References


See Also

PrincipalCoordinates

Examples

data(spiders)

SMACOF

SMACOF

Description

SMACOF algorithm for symmetric proximity matrices

Usage

SMACOF(P, X = NULL, W = NULL, Model = c("Identity", "Ratio", "Interval", "Ordinal"), dimsol = 2, ma
SMACOF performs multidimensional scaling of proximity data to find a least-squares representation of the objects in a low-dimensional space. A majorization algorithm guarantees monotone convergence for optionally transformed, metric and nonmetric data under a variety of models.

Value

An object of class `principal.coordinates` and `MDS`. The function adds the information of the `MDS` to the object of class `proximities`. Together with the information about the proximities the object has:

- **Analysis**
  - The type of analysis performed, "MDS" in this case
- **X**
  - Coordinates for the objects
- **D**
  - Distances
- **Dh**
  - Disparities
- **stress**
  - Raw Stress
- **stress1**
  - stress formula 1
- **stress2**
  - stress formula 2
- **sstress1**
  - sstress formula 1
- **sstress2**
  - sstress formula 2
- **rsq**
  - Squared correlation between disparities and distances
- **rho**
  - Spearman correlation between disparities and distances
- **tau**
  - Kendall correlation between disparities and distances

Author(s)

Jose Luis Vicente-Villardon
References


See Also

MDS, PrincipalCoordinates

Examples

data(spiders)
Dis=BinaryProximities(spiders)
MDSsol=SMACOF(Dis$Proximities)

smoking

Smoking habits

Description

Frequency table representing smoking habits of different employees in a company

Usage

data(smooring)

Format

A data frame with 5 observations on the following 4 variables.

None a numeric vector
Light a numeric vector
Medium a numeric vector
Heavy a numeric vector

Details

Frequency table representing smoking habits of different employees in a company
spiders

Source
http://orange.biolab.si/docs/latest/reference/rst/Orange.projection.correspondence/

References

Examples
data(smoking)
## maybe str(smoking) ; plot(smoking) ...

<table>
<thead>
<tr>
<th>spiders</th>
<th>Hunting Spiders Data</th>
</tr>
</thead>
</table>

Description
Hunting spiders data transformed into Presence/Absence.

Usage
data(spiders)

Format
A data frame with 28 observations of presence/absence of 12 hunting spider species

- **Alopacce** Presence/Absence of the species Alopecosa accentuata
- **Alopcune** Presence/Absence of the species Alopecosa cuneata
- **Alopfabr** Presence/Absence of the species Alopecosa fabrilis
- **Arctlute** Presence/Absence of the species Arctosa lutetiana
- **Arctperi** Presence/Absence of the species Arctosa perita
- **Auloalbi** Presence/Absence of the species Aulonia albimana
- **Pardlugu** Presence/Absence of the species Pardosa lugubris
- **Pardmont** Presence/Absence of the species Pardosa monticola
- **Pardnigr** Presence/Absence of the species Pardosa nigriceps
- **Pardpull** Presence/Absence of the species Pardosa pullata
- **Trocterr** Presence/Absence of the species Trochosa terricola
- **Zoraspin** Presence/Absence of the species Zora spinimana

Source

Examples
data(spiders)
**SpidersEnv**

*Hunting spiders environmental data.*

---

**Description**

Hunting spiders environmental data.

**Usage**

```r
data("SpidersEnv")
```

**Format**

A data frame with 28 observations on the following 6 variables.

- `Watcont` Water content
- `Barsand` Bare sand
- `Covmoss` Cover moss
- `Ligrefl` Light reflection
- `Falltwi` Fallen Twings
- `Coverher` Cover Herbs

**Details**

Hunting spiders environmental data.

**Source**


**References**


**Examples**

```r
data(SpidersEnv)
## maybe str(SpidersEnv) ; plot(SpidersEnv) ...
```
**Description**

Hunting spiders abundances data.

**Usage**

```r
data("SpidersSp")
```

**Format**

A data frame with 28 observations of abundance of 12 hunting spider species

- **Alopacce** Abundance of the species Alopecosa accentuata
- **Alopcune** Abundance of the species Alopecosa cuneata
- **Alopfabr** Abundance of the species Alopecosa fabrilis
- **Arctlute** Abundance of the species Arctosa lutetiana
- **Arctperi** Abundance of the species Arctosa perita
- **Auloalbi** Abundance of the species Aulonia albimana
- **Pardlugu** Abundance of the species Pardosa lugubris
- **Pardmont** Abundance of the species Pardosa monticola
- **Pardnigr** Abundance of the species Pardosa nigriceps
- **Pardpull** Abundance of the species Pardosa pullata
- **Trocterr** Abundance of the species Trochosa terricola
- **Zoraspin** Abundance of the species Zora spinimana

**Source**


**References**


**Examples**

```r
data(SpidersSp)
## maybe str(SpidersSp) ; plot(SpidersSp) ...
```
StatisBiplot

**Description**

The procedure performs STATIS-ACT methodology for multiple tables with common rows and its associated biplot.

**Usage**

```r
StatisBiplot(X, InitTransform = "Standardize columns", dimens = 2,
SameVar = FALSE)
```

**Arguments**

- `X`: A list containing multiple tables with common rows
- `InitTransform`: Initial transformation of the data matrices
- `dimens`: Dimension of the final solution
- `SameVar`: Are the variables the same for all occasions?

**Details**

The procedure performs STATIS-ACT methodology for multiple tables with common rows and its associated biplot. When the variables are the same for all occasions, trajectories for the variables can also be plotted.

**Value**

An object of class StatisBiplot

**Author(s)**

Jose Luis Vicente Villardon

**References**


**Examples**

```r
data(Chemical)
x = Chemical[,5:16]
X=Convert2ThreeWay(x,Chemical$WEEKS, columns=FALSE)
stbip=StatisBiplot(X)
Groups=Chemical$Treatment[1:36]
canstbip=CanonicalStatisBiplot(X, Groups)
```

---

**summary.Canonical.Biplot**

*Summary of the solution of a Canonical Biplot Analysis*

---

**Description**

Summary of the solution of a Canonical Biplot Analysis

**Usage**

```r
## S3 method for class 'Canonical.Biplot'
summary(object, ...)
```

**Arguments**

- `object`
- `...` Additional arguments

**Details**

Summary of the results of a Canonical Biplot

**Value**

The summary

**Author(s)**

Jose Luis Vicente Villardon

**Examples**

```r
### Should be DIRECTLY executable !! ----
```
Summary of the solution of a CCA

Description

Summary of the solution of a CCA

Usage

## S3 method for class 'CCA.sol'
summary(object, ...)

Arguments

object An object of class CCA.sol
...
Aditional arguments

Details

Summary of the solution of a CCA

Value

The main results of a CCA

Author(s)

Jose Luis Vicente Villardon

See Also

CCA

Examples

### Should be DIRECTLY executable !! ----
**summary.MGC**

Arguments

- **object**
  - An object of class "ContinuousBiplot"
- ...
  - Any additional parameters

Details

Summary of the solution of a Biplot for Continuous Data

Value

The summary

Author(s)

Jose Luis Vicente Villardon

Examples

```r
## Simple Biplot with arrows
data(Protein)
bip=PCA.Biplot(Protein[,3:11])
summary(bip)
```

Description

Summarizes the results of Model-Based Gaussian Clustering algorithms

Usage

```r
## S3 method for class 'MGC'
summary(object, Centers = TRUE, Covariances = TRUE, ...)
```

Arguments

- **object**
  - An object of class "MGC"
- **Centers**
  - Should the Centers be shown
- **Covariances**
  - Should the Covariances be shown
- ...
  - Any additional Parameters

Details

Summarizes the results of Model-Based Gaussian Clustering algorithms

Value

No value returned
Author(s)
Jose Luis Vicente Villardon

Examples
### Should be DIRECTLY executable !! ----
### => Define data, use random, 
### or do help(data=index) for the standard data sets.

### The function is currently defined as

summary.Principal.Coordinates

Summary of the results of a Principal Coordinates Analysis

Description
Summary of the results of a Principal Coordinates Analysis

Usage
### S3 method for class 'Principal.Coordinates'
summary(object, printdata=FALSE, printproximities=FALSE, 
printcoordinates=FALSE, printqualities=FALSE,...)

Arguments

object An object of Type Principal.Coordinates
printdata Should original data be printed. Default is FALSE
printproximities Should proximities be printed. Default is FALSE
printcoordinates Should proximities be printed. Default is FALSE
printqualities Should qualities of representation be printed. Default is FALSE
...
Additional parameters to summary.

Details
This function is a method for the generic function summary() for class "Principal.Coordinates". It can be invoked by calling summary(x) for an object x of the appropriate class.

Author(s)
Jose Luis Vicente-Villardon

Examples
data(spiders)
dist=BinaryProximities(spiders)
pco=PrincipalCoordinates(dist)
summary(pco)
**Description**

Plots labels of points in a scattergram. Labels for points with positive $x$ are placed on the right of the points, and labels for points with negative values on the left.

**Usage**

```r
textsmart(A, Labels, CexPoints, ColorPoints, ...)
```

**Arguments**

- `A`: Coordinates of the points for the scattergram
- `Labels`: Labels for the points
- `CexPoints`: Size of the labels
- `ColorPoints`: Colors of the labels
- `...`: Additional graphical arguments

**Details**

The function is used to improve the readability of the labels in a scattergram.

**Value**

No value returned

**Author(s)**

Jose Luis Vicente-Villardon

**See Also**

`plot.Principal.Coordinates`

**Examples**

```r
data(spiders)
dist=BinaryProximities(spiders)
pco=PrincipalCoordinates(dist)
plot(pco, SmartLabels =TRUE)
```
ThreeTwoWay

Converts a multitable list to a two way matrix

Description

Takes a multitable list of matrices X and converts it to a two way matrix with the structure required by the Statis programs using a _ to separate variable and occasion or study.

Usage

ThreeTwoWay(X, whatlines = 2)

Arguments

x

The multitable list.

whatlines

Concatenate the rows (1) or the columns (2)

Details

Takes a multitable list of matrices X and converts it to a two way matrix with the structure required by the Statis programs using a _ to separate variable and occasion or study. When whatlines is 1 the final matrix adds the rows of the three dimensional array, then the columns must be the same for all studies. When whatlines is 2 the columns are concatenated and then the number of rows must be the same for all studies.

Value

A two way matrix

x

A two way matrix

Author(s)

Jose Luis Vicente Villardon

Examples

# No examples yet

TransformIni

Initial transformation of a data matrix

Description

Initial transformation of data before the construction of a biplot. (or any other technique)

Usage

TransformIni(X, transform = "Standardize columns")
Arguments

\( \mathbf{X} \)  
Original Raw Data Matrix

\( \text{transform} \)  
Transformation to use. See details.

Details

Possible Transformations are:

1. "Raw Data": When no transformation is required.
2. "Subtract the global mean": Eliminate an effect common to all the observations.
3. "Double centering": Interaction residuals. When all the elements of the table are comparable. Useful for AMMI models.
4. "Column centering": Remove the column means.
5. "Standardize columns": Remove the column means and divide by its standard deviation.
6. "Row centering": Remove the row means.
7. "Standardize rows": Divide each row by its standard deviation.
8. "Divide by the column means and center": The resulting dispersion is the coefficient of variation.
9. "Normalized residuals from independence" for a contingency table.

The transformation can be provided to the function by using the string between the quotes or just the associated number.

The supplementary rows and columns are not used to calculate the parameters (means, standard deviations, etc). Some of the transformations are not compatible with supplementary data.

Value

\( \mathbf{X} \)  
Transformed data matrix

Author(s)

Jose Luis Vicente Villardon

References


Examples

data(iris)
x=as.matrix(iris[,1:4])
x=TransformIni(x, transform=4)
x
Unfolding

Description

Unfolding para vegetacion

Usage

Unfolding(A, ENV = NULL, TransAbund = "Gaussian", offset = 0.5, weight = "All_1", Constrained = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>The original proximities matrix</td>
</tr>
<tr>
<td>ENV</td>
<td>The matrix of environmental variables</td>
</tr>
<tr>
<td>TransAbund</td>
<td>Initial transformation of the abundances: &quot;None&quot;, &quot;Gaussian&quot;, &quot;Column Percent&quot;, &quot;Gaussian Columns&quot;, &quot;Inverse Square Root&quot;, &quot;Divide by Column Maximum&quot;)</td>
</tr>
<tr>
<td>offset</td>
<td>offset is the quantity added to the zeros of the table</td>
</tr>
<tr>
<td>weight</td>
<td></td>
</tr>
<tr>
<td>Constrained</td>
<td></td>
</tr>
<tr>
<td>TransEnv</td>
<td></td>
</tr>
<tr>
<td>InitConfig</td>
<td></td>
</tr>
<tr>
<td>model</td>
<td></td>
</tr>
<tr>
<td>condition</td>
<td></td>
</tr>
<tr>
<td>r</td>
<td></td>
</tr>
<tr>
<td>maxiter</td>
<td></td>
</tr>
<tr>
<td>tolerance</td>
<td></td>
</tr>
<tr>
<td>lambda</td>
<td></td>
</tr>
<tr>
<td>omega</td>
<td></td>
</tr>
<tr>
<td>plot</td>
<td></td>
</tr>
</tbody>
</table>

Value

An object of class "Unfolding"

Author(s)

Jose Luis Vicente Villardon

References

Ver Articulos

Examples

data("SpidersSp")
unf=Unfolding(SpidersSp)
plot(unf)
GeneFold(SpidersSp)
**VarBiplot**

*Draws a variable on a biplot*

**Description**

Draws a continuous variable on a biplot

**Usage**

```r
VarBiplot(bi1, bi2, b0 = 0, xmin = -3, xmax = 3, ymin = -3, ymax = 3, label = "Point", mode = "a", CexPoint = 0.8,
           PchPoint = 1, Color = "blue", ticks = c(-3, -2.5, -2, -1.5, -1, -0.5, 0.5, 1, 1.5, 2, 2.5, 3), ticklabels =
           round(ticks, digits = 2), tl = 0.03, ts = "Complete",
           Position = "Angle", AddArrow=FALSE, ...)
```

**Arguments**

- `bi1`: First component of the direction vector
- `bi2`: Second component of the direction vector
- `b0`: Constant for the regression adjusted biplots
- `xmin`: Minimum value of the x axis
- `xmax`: Maximum value of the x axis
- `ymin`: Minimum value of the y axis
- `ymax`: Maximum value of the y axis
- `label`: Label of the variable
- `mode`: Mode of the biplot: "p", "a", "b", "h", "ah" and "s".
- `CexPoint`: Size for the symbols and labels of the variables
- `PchPoint`: Symbols for the variable (when represented as a point)
- `Color`: Color for the variable
- `ticks`: Ticks when the variable is represented as a graded scale
- `ticklabels`: Labels for the ticks when the variable is represented as a graded scale
- `tl`: Thick length
- `ts`: Size of the mark in the graded scale
- `Position`: If the Position is "Angle" the label of the variable is placed using the angle of the vector
- `AddArrow`: Add an arrow to the representation of other modes of the biplot.
- `...`: Any other graphical parameters

**Details**

See plot.PCA.Biplot

**Value**

No value returned
Extracts the weighted averages of a CCA solution

Description

Extracts the weighted averages of a CCA solution

Usage

wa(CCA.sol, transformed = FALSE)

Arguments

CCA.sol The solution of a CCA
transformed Average of the transformed or the original data?

Details

Extracts the weighted averages of a CCA solution

Value

A matrix with the averages

Author(s)

Jose Luis Vicente Villardon

Examples

# Should be DIRECTLY executable !!!

data(Protein)
bip=PCA.Biplot(Protein[,3:11])
plot(bip)
**wcor**  
*Weighted correlations*

**Description**
Weighted correlations

**Usage**
\[
\text{wcor}(d1, d2, w = \text{rep}(1, nrow(d1))/nrow(d1))
\]

**Arguments**
- \text{d1}: First Vector
- \text{d2}: Second vector to correlate
- \text{w}: weights for each element of the vectors

**Details**
Weighted correlations

**Value**
Weighted correlation

**Author(s)**
Jose Luis Vicente Villardon

---

**weighted.quantile**  
*Weighted quantiles*

**Description**
Weighted quantiles

**Usage**
\[
\text{weighted.quantile}(x, w, q = 0.5)
\]

**Arguments**
- \text{x}: The numerical variable.
- \text{w}: Weights
- \text{q}: Quantile

**Value**
The quantile
WeightedPCoA

Author(s)
Jose Luis Vicente Villardon

Examples
### Should be DIRECTLY executable !! ----

| WeightedPCoA | Weighted Principal Coordinates Analysis |

Description
Weighted Principal Coordinates Analysis

Usage
WeightedPCoA(Proximities, weigths = matrix(1,dim(Proximities)$Proximities)[1],1), dimension = 2,

Arguments

- **Proximities**: A matrix containing the proximities among a set of objects
- **weigths**: Weights
- **dimension**: Dimension of the solution
- **tolerance**: Tolerance for the eigenvalues

Details
Weighted Principal Coordinates Analysis

Value
data(spiders) dist=BinaryProximities(spiders) pco=WeightedPCoA(dist) An object of class Principal.Coordinates

Author(s)
Jose Luis Vicente-Villardon

References
See Also

BinaryProximities

Examples

data(spiders)
dist=BinaryProximities(spiders)
pco=WeightedPCoA(dist)

---

Wine data

Description

Comparison of young wines of Ribera de Duero and Toro

Usage

data("wine")

Format

A data frame with 45 observations on the following 21 variables.

- Year: a factor with levels 1986 1987
- Origin: a factor with levels Ribera Toro
- Group: a factor with levels R86 R87 T86 T87
- A: a numeric vector
- VA: a numeric vector
- TA: a numeric vector
- FA: a numeric vector
- pH: a numeric vector
- TPR: a numeric vector
- TPS: a numeric vector
- V: a numeric vector
- PC: a numeric vector
- ACR: a numeric vector
- ACS: a numeric vector
- ACC: a numeric vector
- CI: a numeric vector
- CI2: a numeric vector
- H: a numeric vector
- I: a numeric vector
- CA: a numeric vector
- VPC: a numeric vector
**Details**

Comparison of young wines of Ribera de Duero and Toro

**Source**


**References**


**Examples**

```r
data(wine)
# maybe str(wine) ; plot(wine) ...
```

---

**Description**

Matrix of zeros

**Usage**

```r
zeros(n)
```

**Arguments**

- `n` Dimension of the matrix

**Value**

A matrix of zeros

**Author(s)**

Jose Luis Vicente Villardon

**Examples**

```r
zeros(6)
```
Index

*Topic 3D Biplot
  plot3dContinuousBiplot, 108

*Topic Alternated Least Squares
  CrissCross, 33

*Topic Binary Data Matrix
  CheckBinaryMatrix, 26

*Topic Biplot
  AddCluster2Biplot, 5

*Topic Canonical Variate Analysis
  plotCanonicalBiplot, 87

*Topic Cluster Analysis
  AddCluster2Biplot, 5

*Topic Discriminant Analysis
  plotCanonicalBiplot, 87

*Topic Distance
  NominalDistances, 75

*Topic EM
  OrdLogBipEM, 79

*Topic MANOVA
  plotCanonicalBiplot, 87

*Topic Multivariate
  NominalDistances, 75

*Topic Nominal
  NominalDistances, 75

*Topic \textasciitildeBiplot
  CrissCross, 33

*Topic \textasciitildeBiplotkw1
  BootstrapDistance, 12
  BootstrapScalar, 14
  CheckBinaryMatrix, 26
  Dataframe2BinaryMatrix, 36
  DimensionLabels, 40
dlines, 40
  Factor2Binary, 48
  LogFrequencyBiplot, 61
  logit, 63
  plotContinuousBiplot, 92
  plotEllipse, 95
  plotUnfolding, 107
  PlotOrdinalResponses, 113
  PLSRfit, 114
  RidgeBinaryLogistic, 122
  VarBiplot, 149

*Topic \textasciitildeBiplotkw2
  OrdinalLogisticFit, 78
  RidgeMultinomialLogisticFit, 126
  RidgeMultinomialLogisticRegression, 128
  RidgeOrdinalLogistic, 129

*Topic \textasciitildekwd2
  Chemical, 28
  Doctors, 41
  moth, 71
  PoliticalFigures, 115
  Protein, 119
  RAPD, 120
  riano, 121
  smoking, 136
  spiders, 137
  SpidersEnv, 138
  SpidersSp, 139
  wine, 153

*Topic logistic
  OrdinalLogisticFit, 78
  RidgeMultinomialLogisticFit, 126
  RidgeMultinomialLogisticRegression, 128
  RidgeOrdinalLogistic, 129

*Topic models
  OrdinalLogisticFit, 78
  RidgeMultinomialLogisticFit, 126
  RidgeMultinomialLogisticRegression, 128
  RidgeOrdinalLogistic, 129
INDEX

*Topic **package**
  MultibiplotR-package, 4
*Topic **ridge**
  RidgeMultinomialLogisticFit, 126
*Topic **summary**
  summary.Principal.Coordinates, 144

AddCluster2Biplot, 5, 50, 112
AddContVars2Biplot, 7, 8
AddSupVars2Biplot, 8, 8

BinaryDistances, 9, 12, 76
BinaryProximities, 10, 104, 105, 118, 153
BootstrapDistance, 12, 118
BootstrapScalar, 14, 14, 16, 18
BootstrapSmacof, 16, 68

CA, 18
CanonicalBiplot, 19
CanonicalDistanceAnalysis, 20
CanonicalStatisBiplot, 22
CategoricalDistances, 23
CategoricalProximities, 24
CCA, 25, 131, 142
CheckBinaryMatrix, 26
CheckBinaryVector, 27
Chemical, 28
Circle, 29
ConcEllipse, 29, 50, 96
ContinuousDistances, 30, 76
Convert2ThreeWay, 32
ConvertFactors2Integers, 33
CrissCross, 33, 63
CumSum, 35

Dataframe2BinaryMatrix, 12, 36
DataFrame2Matrix4Regression, 37
DensityBiplot, 37
Dhats, 38
diagonal, 39
DimensionLabels, 40
dlines, 40
Doctors, 41

EuclideanDistance, 42
ExpandTable, 43
ExternalBinaryLogisticBiplot, 43, 98
ExtractTable, 45

FA.Biplot, 46
Factor2Binary, 48
Fraction, 49, 99

GeneralizedProcrustes, 50

GetBiplotScales, 51
GetCCAScales, 52
ginv, 53, 66
GowerProximities, 54
GowerSimilarities, 55

Hermquad, 56
HJ.Biplot, 56

InBox, 58
InitialTransform, 48, 58, 59, 86
Integer2Binary, 60

LogFrequencyBiplot, 35, 61
logit, 63

Matrix2Proximities, 64
matrixsqrt, 65
matrixsqrtinv, 66
MDS, 67, 136
MGC, 69
MonotoneRegression, 70
moth, 71
MultBiplot (MultBiplotR-package), 4
MultBiplotR-package, 4
Multiquad, 72
MultiTableStatistics, 72
MultiTableTransform, 73

NiceNumber, 74, 117
NominalDistances, 75
Numeric2Binary, 76

ones, 77
OrdinalLogisticFit, 78
OrdLogBipEM, 79
OrdVarBiplot, 81
OrdVarCoordinates, 82
OrthogonalizeScores, 83

PCA.Biplot, 84
plot.CA.sol, 86
plot.Canonical.Biplot, 87, 111
plot.CCA.sol, 90
plot.ContinuousBiplot, 87, 91, 92, 102, 110, 150
plot.ellipse, 95
plot.fraction, 98
plot.MGC, 99
plot.Ordinal.Logistic.Biplot, 100
plot.PCoABootstrap, 102
plot.Principal.Coordinates, 103, 145
INDEX

plot.Procrustes, 105
plot.Statis, 106
plot.Unfolding, 107
plot3d.ContinuousBiplot, 108
plot3dCanonicalBiplot, 110
PlotBiplotClusters, 112
PlotOrdinalResponses, 113
PLSRfit, 114
PoliticalFigures, 115
PrettyTicks, 74, 116
PrincipalCoordinates, 10, 31, 51, 117, 134, 136
print.MGC, 118
Protein, 119

RAPD, 120
RemoveRowsWithNaNs, 121
Riano, 121
RidgeBinaryLogistic, 122, 126
RidgeBinaryLogisticFit, 125
RidgeMultinomialLogisticFit, 126, 129
RidgeMultinomialLogisticRegression, 128
RidgeOrdinalLogistic, 129

scores.CCA.sol, 131
SelectvarsAnova, 22
SeparateVarTypes, 132
SimpleProcrustes, 133
SMACOF, 134
smoking, 136
spiders, 137
SpidersEnv, 138
SpidersSp, 139
StatisBiplot, 140
summary.Canonical.Biplot, 141
summary.CCA.sol, 142
summary.ContinuousBiplot, 142
summary.MGC, 143
summary.PrincipalCoordinates, 144

textsmart, 145
Three2TwoWay, 146
TransformIni, 146

Unfolding, 148

VarBiplot, 149

wa, 150
wcor, 151
weighted.quantile, 151
WeightedPCoA, 152
wine, 153

zeros, 154