

Package ‘TExPosition’

January 20, 2025

Type Package

Title Two-Table ExPosition

Version 2.6.10.1

Date 2013-12-09

Author Derek Beaton, Jenny Rieck, Cherise R. Chin Fatt, Herve Abdi

Maintainer Derek Beaton <exposition.software@gmail.com>

Description An extension of ExPosition for two table analyses, specifically, discriminant analyses.

License GPL-2

Depends prettyGraphs (>= 2.1.4), ExPosition (>= 2.0.0)

BugReports <http://code.google.com/p/exposition-family/issues/list>

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-31 17:27:48 UTC

Contents

TExPosition-package	2
calculateLVConstraints	3
fastEucCalc	4
fii2fi	4
print.tepAssign	5
print.tepBADA	6
print.tepDICA	6
print.tepGPLS	7
print.tepGraphs	7
print.tepPLS	8
print.tepPLSCA	8
print.texpoOutput	9
R2	9
tepBADA	10
tepDICA	11

tepGPLS	13
tepGraphs	15
tepPLS	17
tepPLSCA	18
texpoDesignCheck	20

Index	22
--------------	-----------

TExPosition-package *TExPosition: Two-table analyses with via ExPosition.*

Description

TExPosition is two-table [ExPosition](#) and includes discriminant methods of the singular value decomposition (SVD). The core of TExPosition is [ExPosition](#) and the [svd](#).

Details

Package: TExPosition
 Type: Package
 Version: 2.6.10
 Date: 2013-12-00
 Depends: R (>=2.15.0), prettyGraphs (>= 2.1.4), ExPosition (>= 2.0.0)
 License: GPL-2
 URL: <http://www.utdallas.edu/~derekbeaton/software/ExPosition>

Author(s)

Questions, comments, compliments, and complaints go to Derek Beaton <exposition.software@gmail.com>.

The following people are authors or contributors to TExPosition code, data, or examples:
 Derek Beaton, Jenny Rieck, Cherise Chin-Fatt, Francesca Filbey, and Hervé Abdi.

References

- Abdi, H., and Williams, L.J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2, 433-459.
- Abdi, H. and Williams, L.J. (2010). Correspondence analysis. In N.J. Salkind, D.M., Dougherty, & B. Frey (Eds.): *Encyclopedia of Research Design*. Thousand Oaks (CA): Sage. pp. 267-278.
- Abdi, H. (2007). Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD). In N.J. Salkind (Ed.): *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): Sage. pp. 907-912.
- Abdi, H. & Williams, L.J. (2010). Barycentric discriminant analysis (BADIA). In N.J. Salkind,

D.M., Dougherty, & B. Frey (Eds.): *Encyclopedia of Research Design*. Thousand Oaks (CA): Sage. pp. 64-75.

Abdi, H. (2007). Discriminant correspondence analysis. In N.J. Salkind (Ed.): *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): Sage. pp. 270-275. Krishnan, A., Williams, L. J., McIntosh, A. R., & Abdi, H. (2011). Partial Least Squares (PLS) methods for neuroimaging: A tutorial and review. *NeuroImage*, 56(2), 455 – 475.

McIntosh, A. R., & Lobaugh, N. J. (2004). Partial least squares analysis of neuroimaging data: applications and advances. *Neuroimage*, 23, S250–S263.

See Also

[tepBADA](#), [tepPLS](#), [tepGPLS](#), [tepDICA](#), [tepPLSCA](#)

Examples

#For more examples, see each individual function (as noted above).

calculateLVConstraints

calculateLVConstraints

Description

Calculates constraints for plotting latent variables.

Usage

```
calculateLVConstraints(results,x_axis=1,y_axis=2,constraints=NULL)
```

Arguments

results	results (with \$lx and \$ly) from TExPosition (i.e., \$TExPosition.Data)
x_axis	which component should be on the x axis?
y_axis	which component should be on the y axis?
constraints	if available, axis constraints for the plots (determines end points of the plots).

Value

Returns a list with the following items:

\$constraints	axis constraints for the plots (determines end points of the plots).
---------------	----------------------------------------------------------------------

Author(s)

Derek Beaton

 fastEucCalc

fastEucCalc

Description

Fast Euclidean distance calculations.

Usage

```
fastEucCalc(x, c)
```

Arguments

x	a set of points.
c	a set of centers.

Details

This function is especially useful for discriminant analyses. The distance from each point in *x* to each point in *c* is computed and returned as a `nrow(x) x nrow(c)` matrix.

Value

a distance matrix
Euclidean distances of each point to each center are returned.

Author(s)

Hervé Abdi, Derek Beaton

fii2fi

fii2fi: individuals to centers

Description

All computations between individual factor scores (*fii*) and group factor scores (*fi*).

Usage

```
fii2fi(DSIGN, fii, fi)
```

Arguments

DESIGN	a dummy-coded design matrix
fii	a set of factor scores for individuals (rows)
fi	a set of factor scores for rows

Value

A list of values containing:

distances	Euclidean distances of all rows to each category center
assignments	an assignment matrix (similar to DESIGN) where each individual is assigned to the closest category center
confusion	a confusion matrix of how many items are assigned (and mis-assigned) to each category

Author(s)

Hervé Abdi, Derek Beaton

`print.tepAssign` *Print assignment results*

Description

Print assignment results.

Usage

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

Arguments

x an list that contains items to make into the tepAssign class.
... inherited/passed arguments for S3 print method(s).

Author(s)

Derek Beaton, Cherise Chin-Fatt

<code>print.tepBADA</code>	<i>Print tepBADA results</i>
----------------------------	------------------------------

Description

Print tepBADA results.

Usage

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

Arguments

<code>x</code>	an list that contains items to make into the tepBADA class.
<code>...</code>	inherited/passed arguments for S3 print method(s).

Author(s)

Derek Beaton, Cherise Chin-Fatt

<code>print.tepDICA</code>	<i>Print tepDICA results</i>
----------------------------	------------------------------

Description

Print tepDICA results.

Usage

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

Arguments

<code>x</code>	an list that contains items to make into the tepDICA class.
<code>...</code>	inherited/passed arguments for S3 print method(s).

Author(s)

Derek Beaton, Cherise Chin-Fatt

print.tepGPLS *Print tepGPLS results*

Description

Print tepGPLS results.

Usage

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

Arguments

x an list that contains items to make into the tepGPLS class.
... inherited/passed arguments for S3 print method(s).

Author(s)

Derek Beaton, Cherise Chin-Fatt

print.tepGraphs *Print tepGraphs results*

Description

Print tepGraphs results.

Usage

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

Arguments

x an list that contains items to make into the tepGraphs class.
... inherited/passed arguments for S3 print method(s).

Author(s)

Derek Beaton, Cherise Chin-Fatt

print.tepPLS *Print tepPLS results*

Description

Print tepPLS results.

Usage

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

Arguments

x an list that contains items to make into the tepPLS class.
... inherited/passed arguments for S3 print method(s).

Author(s)

Derek Beaton, Cherise Chin-Fatt

print.tepPLSCA *Print tepPLSCA results*

Description

Print tepPLSCA results.

Usage

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

Arguments

x an list that contains items to make into the tepPLSCA class.
... inherited/passed arguments for S3 print method(s).

Author(s)

Derek Beaton, Cherise Chin-Fatt

print.texpoOutput *Print TExPosition results*

Description

Print TExPosition results.

Usage

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

Arguments

x an list that contains items to make into the texpoOutput class.
... inherited/passed arguments for S3 print method(s).

Author(s)

Derek Beaton, Cherise Chin-Fatt

R2 *R-squared computations*

Description

A function to compute R-squared for BADA and DICA

Usage

```
R2(group.masses, di, ind.masses = NULL, dii)
```

Arguments

group.masses a masses matrix for the groups
di a set of squared distances of the groups
ind.masses a masses matrix for the individuals
dii a set of squared distances for the individuals

Value

R2 An R-squared

Author(s)

Jenny Rieck, Derek Beaton

 tepBADA

Barycentric Discriminant Analysis

Description

Barycentric Discriminant Analysis (BADA) via TExPosition.

Usage

```
tepBADA(DATA, scale = TRUE, center = TRUE, DESIGN = NULL, make_design_nominal = TRUE,
group.masses = NULL, weights = NULL, graphs = TRUE, k = 0)
```

Arguments

DATA	original data to perform a BADA on.
scale	a boolean, vector, or string. See expo.scale for details.
center	a boolean, vector, or string. See expo.scale for details.
DESIGN	a design matrix to indicate if rows belong to groups. Required for BADA.
make_design_nominal	a boolean. If TRUE (default), DESIGN is a vector that indicates groups (and will be dummy-coded). If FALSE, DESIGN is a dummy-coded matrix.
group.masses	a diagonal matrix or column-vector of masses for the groups.
weights	a diagonal matrix or column-vector of weights for the column items.
graphs	a boolean. If TRUE (default), graphs and plots are provided (via tepGraphs)
k	number of components to return.

Details

Note: BADA is a special case of PLS ([tepPLS](#), [tepGPLS](#)) wherein DATA1 are data and DATA2 are a group-coded disjunctive matrix. This is also called mean-centered PLS (Krishnan et al., 2011).

Value

See [epGPCA](#) (and also [corePCA](#)) for details on what is returned. In addition to the values returned:

fii	factor scores computed for supplemental observations
dii	squared distances for supplemental observations
rii	cosines for supplemental observations
assign	a list of assignment data. See fii2fi and R2
lx	latent variables from DATA1 computed for observations
ly	latent variables from DATA2 computed for observations

Author(s)

Derek Beaton

References

- Abdi, H., and Williams, L.J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2, 433-459.
- Abdi, H. and Williams, L.J. (2010). Correspondence analysis. In N.J. Salkind, D.M., Dougherty, & B. Frey (Eds.): *Encyclopedia of Research Design*. Thousand Oaks (CA): Sage. pp. 267-278.
- Abdi, H. (2007). Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD). In N.J. Salkind (Ed.): *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): Sage. pp. 907-912.
- Abdi, H. & Williams, L.J. (2010). Barycentric discriminant analysis (BADIA). In N.J. Salkind, D.M., Dougherty, & B. Frey (Eds.): *Encyclopedia of Research Design*. Thousand Oaks (CA): Sage. pp. 64-75.
- Abdi, H., Williams, L.J., Beaton, D., Posamentier, M., Harris, T.S., Krishnan, A., & Devous, M.D. (in press, 2012). Analysis of regional cerebral blood flow data to discriminate among Alzheimer's disease, fronto-temporal dementia, and elderly controls: A multi-block barycentric discriminant analysis (MUBADA) methodology. *Journal of Alzheimer Disease*, , -. Abdi, H., Williams, L.J., Connolly, A.C., Gobbin, M.I., Dunlop, J.P., & Haxby, J.V. (2012). Multiple Subject Barycentric Discriminant Analysis (MUSUBADA): How to assign scans to categories without using spatial normalization. *Computational and Mathematical Methods in Medicine*, 2012, 1-15. doi:10.1155/2012/634165.
- Krishnan, A., Williams, L. J., McIntosh, A. R., & Abdi, H. (2011). Partial Least Squares (PLS) methods for neuroimaging: A tutorial and review. *NeuroImage*, 56(2), 455 – 475.

See Also

[corePCA](#), [epPCA](#), [epGPCA](#), [epMDS](#)

For MatLab code: http://utd.edu/~derekbeaton/attachments/Software/matlab/MuSuBADA_V3.zip

Examples

```
data(bada.wine)
bada.res <- tepBADA(bada.wine$data, scale=FALSE, DESIGN=bada.wine$design, make_design_nominal=FALSE)
```

 tepDICA

Discriminant Correspondence Analysis

Description

Discriminant Correspondence Analysis (DICA) via TExPosition.

Usage

```
tepDICA(DATA, make_data_nominal = FALSE, DESIGN = NULL, make_design_nominal = TRUE,
group.masses = NULL, weights = NULL, symmetric = TRUE, graphs = TRUE, k = 0)
```

Arguments

DATA	original data to perform a DICA on. Data can be contingency (like CA) or categorical (like MCA).
make_data_nominal	a boolean. If TRUE (default), DATA is recoded as a dummy-coded matrix. If FALSE, DATA is a dummy-coded matrix.
DESIGN	a design matrix to indicate if rows belong to groups. Required for DICA.
make_design_nominal	a boolean. If TRUE (default), DESIGN is a vector that indicates groups (and will be dummy-coded). If FALSE, DESIGN is a dummy-coded matrix.
group.masses	a diagonal matrix or column-vector of masses for the groups.
weights	a diagonal matrix or column-vector of weights for the column it
symmetric	a boolean. If TRUE (default) symmetric factor scores for rows.
graphs	a boolean. If TRUE (default), graphs and plots are provided (via tepGraphs)
k	number of components to return.

Details

If you use Hellinger distance, it is best to set `symmetric` to FALSE.

Note: DICA is a special case of PLS-CA ([tepPLSCA](#)) wherein DATA1 are data and DATA2 are a group-coded disjunctive matrix.

Value

See [epCA](#) (and also [coreCA](#)) for details on what is returned. In addition to the values returned:

fii	factor scores computed for supplemental observations
dii	squared distances for supplemental observations
rii	cosines for supplemental observations
assign	a list of assignment data. See fii2fi and R2
lx	latent variables from DATA1 computed for observations
ly	latent variables from DATA2 computed for observations

Author(s)

Derek Beaton, Hervé Abdi

References

- Abdi, H., and Williams, L.J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2, 433-459.
- Abdi, H. and Williams, L.J. (2010). Correspondence analysis. In N.J. Salkind, D.M., Dougherty, & B. Frey (Eds.): *Encyclopedia of Research Design*. Thousand Oaks (CA): Sage. pp. 267-278.

Abdi, H. (2007). Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD). In N.J. Salkind (Ed.): *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): Sage. pp. 907-912.

Abdi, H. (2007). Discriminant correspondence analysis. In N.J. Salkind (Ed.): *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): Sage. pp. 270-275.

Pinkham, A.E., Sasson, N.J., Beaton, D., Abdi, H., Kohler, C.G., Penn, D.L. (in press, 2012). Qualitatively distinct factors contribute to elevated rates of paranoia in autism and schizophrenia. *Journal of Abnormal Psychology*, 121, -.

Williams, L.J., Abdi, H., French, R., & Orange, J.B. (2010). A tutorial on Multi-Block Discriminant Correspondence Analysis (MUDICA): A new method for analyzing discourse data from clinical populations. *Journal of Speech Language and Hearing Research*, 53, 1372-1393.

Williams, L.J., Dunlop, J.P., & Abdi, H. (2012). Effect of age on the variability in the production of text-based global inferences. *PLoS One*, 7(5): e36161. doi:10.1371/journal.pone.0036161 (pp.1-9)

See Also

[coreCA](#), [epCA](#), [epMCA](#)

For MatLab code: http://utd.edu/~herve/HerveAbdi_MatlabPrograms4MUDICA.zip For additional R code (with inference tests): <http://utdallas.edu/~dfb090020/attachments/MuDiCA.zip>

Examples

```
data(dica.wine)
dica.res <- tepDICA(dica.wine$data,DESIGN=dica.wine$design,make_design_nominal=FALSE)
```

tepGPLS

Generalized Partial Least Squares

Description

Generalized Partial Least Squares (GPLS) via TExPosition. GPLS is to PLS ([tepPLS](#)) as PCA [epPCA](#) is to GPCA [epGPCA](#). The major difference between PLS and GPLS is that GPLS allows the use of weights for the columns of each data set (just like GPCA).

Usage

```
tepGPLS(DATA1, DATA2,
center1 = TRUE, scale1 = "SS1",
center2 = TRUE, scale2 = "SS1",
DESIGN = NULL, make_design_nominal = TRUE,
weights1 = NULL, weights2 = NULL,
graphs = TRUE, k = 0)
```

Arguments

DATA1	Data matrix 1 (X)
DATA2	Data matrix 2 (Y)

center1	a boolean, vector, or string to center DATA1. See expo.scale for details.
scale1	a boolean, vector, or string to scale DATA1. See expo.scale for details.
center2	a boolean, vector, or string to center DATA2. See expo.scale for details.
scale2	a boolean, vector, or string to scale DATA2. See expo.scale for details.
DESIGN	a design matrix to indicate if rows belong to groups.
make_design_nominal	a boolean. If TRUE (default), DESIGN is a vector that indicates groups (and will be dummy-coded). If FALSE, DESIGN is a dummy-coded matrix.
weights1	a weight vector (or diag matrix) for the columns of DATA1.
weights2	a weight vector (or diag matrix) for the columns of DATA2.
graphs	a boolean. If TRUE (default), graphs and plots are provided (via tepGraphs)
k	number of components to return.

Details

This implementation of Partial Least Squares is a symmetric analysis. It was first described by Tucker (1958), again by Bookstein (1994), and has gained notoriety in Neuroimaging from McIntosh et al., (1996). This particular implementation allows the user to provide weights for the columns of both DATA1 and DATA2.

Value

See [epGPCA](#) (and also [corePCA](#)) for details on what is returned. In addition to the values returned:

lx	latent variables from DATA1 computed for observations
ly	latent variables from DATA2 computed for observations
data1.norm	center and scale information for DATA1
data2.norm	center and scale information for DATA2

Author(s)

Derek Beaton

References

- Tucker, L. R. (1958). An inter-battery method of factor analysis. *Psychometrika*, 23(2), 111–136.
- Bookstein, F., (1994). Partial least squares: a dose–response model for measurement in the behavioral and brain sciences. *Psychology* 5 (23)
- Abdi, H. (2007). Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD). In N.J. Salkind (Ed.): *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): Sage. pp. 907-912.
- Krishnan, A., Williams, L. J., McIntosh, A. R., & Abdi, H. (2011). Partial Least Squares (PLS) methods for neuroimaging: A tutorial and review. *NeuroImage*, 56(2), 455 – 475.
- McIntosh, A. R., & Lobaugh, N. J. (2004). Partial least squares analysis of neuroimaging data: applications and advances. *Neuroimage*, 23, S250–S263.

See Also

[corePCA](#), [epPCA](#), [epGPCA](#), [tepPLS](#), [tepPLSCA](#), [tepBADA](#), [tepDICA](#)

Examples

```
data(beer.tasting.notes)
data1<-beer.tasting.notes$data[,1:8]
data2<-beer.tasting.notes$data[,9:16]
gpls.res <- tepGPLS(data1,data2)
```

 tepGraphs

tepGraphs: TExPosition plotting function

Description

TExPosition plotting function which is an interface to [prettyGraphs](#).

Usage

```
tepGraphs(res, x_axis = 1, y_axis = 2,
  tepPlotInfo = NULL, DESIGN = NULL,
  fi.col = NULL, fi.pch = NULL, fii.col = NULL, fii.pch = NULL,
  fj.col = NULL, fj.pch = NULL, col.offset = NULL,
  constraints = NULL, lv.constraints = NULL,
  xlab = NULL, ylab = NULL, main = NULL,
  lvPlots = TRUE, lvAgainst = TRUE,
  contributionPlots = TRUE, correlationPlotter = TRUE,
  showHulls = 1, biplots = FALSE, graphs = TRUE)
```

Arguments

<code>res</code>	results from TExPosition
<code>x_axis</code>	which component should be on the x axis?
<code>y_axis</code>	which component should be on the y axis?
<code>tepPlotInfo</code>	A list (<code>\$Plotting.Data</code>) from <code>tepGraphs</code> or <code>TExPosition</code> .
<code>DESIGN</code>	A design matrix to apply colors (by pallete selection) to row items
<code>fi.col</code>	A matrix of colors for the group items. If <code>NULL</code> , colors will be selected.
<code>fi.pch</code>	A matrix of pch values for the group items. If <code>NULL</code> , pch values are all 21.
<code>fii.col</code>	A matrix of colors for the row items (observations). If <code>NULL</code> , colors will be selected.
<code>fii.pch</code>	A matrix of pch values for the row items (observations). If <code>NULL</code> , pch values are all 21.
<code>fj.col</code>	A matrix of colors for the column items. If <code>NULL</code> , colors will be selected.
<code>fj.pch</code>	A matrix of pch values for the column items. If <code>NULL</code> , pch values are all 21.

col.offset	A numeric offset value. Is passed to createColorVectorsByDesign .
constraints	Plot constraints as returned from prettyPlot . If NULL, constraints are selected.
lv.constraints	Plot constraints for latent variables. If NULL, constraints are selected.
xlab	x axis label
ylab	y axis label
main	main label for the graph window
lvPlots	a boolean. If TRUE, latent variables (X, Y) are plotted. If FALSE, latent variables are not plotted.
lvAgainst	a boolean. If TRUE, latent variables (X, Y) are plotted against each other. If FALSE, latent variables are plotted like factor scores.
contributionPlots	a boolean. If TRUE (default), contribution bar plots will be created.
correlationPlotter	a boolean. If TRUE (default), a correlation circle plot will be created. Applies to PCA family of methods (CA is excluded for now).
showHulls	a value between 0 and 1 to make a peeled hull at that percentage. All values outside of 0-1 will not plot any hulls.
biplots	a boolean. If FALSE (default), separate plots are made for row items (\$fii and \$fi) and column items (\$fj). If TRUE, row (\$fii and \$fi) and column (\$fj) items will be on the same plot.
graphs	a boolean. If TRUE, graphs are created. If FALSE, only data associated to plotting (e.g., constraints, colors) are returned.

Details

tepGraphs is an interface between [TExPosition](#) and [prettyGraphs](#).

Value

The following items are bundled inside of `$Plotting.Data`:

<code>\$fii.col</code>	the colors that are associated to the individuals (row items; \$fii).
<code>\$fii.pch</code>	the pch values associated to the individuals (row items; \$fii).
<code>\$fi.col</code>	the colors that are associated to the groups (\$fi).
<code>\$fi.pch</code>	the pch values associated to the groups (\$fi).
<code>\$fj.col</code>	the colors that are associated to the column items (\$fj).
<code>\$fj.pch</code>	the pch values associated to the column items (\$fj).
<code>\$constraints</code>	axis constraints for the plots (determines end points of the plots).

Author(s)

Derek Beaton

See Also[prettyGraphs](#)**Examples**

```
#this is for TExPosition's iris data
data(ep.iris)
bada.iris <- tepBADA(ep.iris$data,DESIGN=ep.iris$design,make_design_nominal=FALSE)
#there are only 2 components, not 3.
bada.iris.plotting.data.biplot <- tepGraphs(bada.iris,x_axis=1,y_axis=2,biplots=TRUE)
```

tepPLS

*Partial Least Squares***Description**

Partial Least Squares (PLS) via TExPosition.

Usage

```
tepPLS(DATA1, DATA2,
center1 = TRUE, scale1 = "SS1", center2 = TRUE, scale2 = "SS1",
DESIGN = NULL, make_design_nominal = TRUE,
graphs = TRUE, k = 0)
```

Arguments

DATA1	Data matrix 1 (X)
DATA2	Data matrix 2 (Y)
center1	a boolean, vector, or string to center DATA1. See expo.scale for details.
scale1	a boolean, vector, or string to scale DATA1. See expo.scale for details.
center2	a boolean, vector, or string to center DATA2. See expo.scale for details.
scale2	a boolean, vector, or string to scale DATA2. See expo.scale for details.
DESIGN	a design matrix to indicate if rows belong to groups.
make_design_nominal	a boolean. If TRUE (default), DESIGN is a vector that indicates groups (and will be dummy-coded). If FALSE, DESIGN is a dummy-coded matrix.
graphs	a boolean. If TRUE (default), graphs and plots are provided (via tepGraphs)
k	number of components to return.

Details

This implementation of Partial Least Squares is a symmetric analysis. It was first described by Tucker (1958), again by Bookstein (1994), and has gained notoriety in Neuroimaging from McIntosh et al., (1996).

Value

See [epGPCA](#) (and also [corePCA](#)) for details on what is returned. In addition to the values returned:

lx	latent variables from DATA1 computed for observations
ly	latent variables from DATA2 computed for observations
data1.norm	center and scale information for DATA1
data1.norm	center and scale information for DATA2

Author(s)

Derek Beaton

References

- Tucker, L. R. (1958). An inter-battery method of factor analysis. *Psychometrika*, 23(2), 111–136.
- Bookstein, F., (1994). Partial least squares: a dose–response model for measurement in the behavioral and brain sciences. *Psychology* 5 (23)
- McIntosh, A. R., Bookstein, F. L., Haxby, J. V., & Grady, C. L. (1996). Spatial Pattern Analysis of Functional Brain Images Using Partial Least Squares. *NeuroImage*, 3(3), 143–157.
- Krishnan, A., Williams, L. J., McIntosh, A. R., & Abdi, H. (2011). Partial Least Squares (PLS) methods for neuroimaging: A tutorial and review. *NeuroImage*, 56(2), 455 – 475.
- McIntosh, A. R., & Lobaugh, N. J. (2004). Partial least squares analysis of neuroimaging data: applications and advances. *Neuroimage*, 23, S250–S263.

See Also

[corePCA](#), [epPCA](#), [epGPCA](#), [tepBADA](#), [tepGPLS](#), [tepPLSCA](#)

Examples

```
data(beer.tasting.notes)
data1<-beer.tasting.notes$data[,1:8]
data2<-beer.tasting.notes$data[,9:16]
pls.res <- tepPLS(data1,data2)
```

 tepPLSCA

Partial Least Squares-Correspondence Analysis

Description

Partial Least Squares-Correspondence Analysis (PLSCA) via TExPosition.

Usage

```
tepPLSCA(DATA1, DATA2, make_data1_nominal = FALSE, make_data2_nominal = FALSE,
DESIGN = NULL, make_design_nominal = TRUE,
weights1=NULL, weights2 = NULL,
symmetric = TRUE, graphs = TRUE, k = 0)
```

Arguments

DATA1	Data matrix 1 (X), must be categorical (like MCA) or in disjunctive code see <code>make_data1_nominal</code> .
DATA2	Data matrix 2 (Y), must be categorical (like MCA) or in disjunctive code see <code>make_data2_nominal</code> .
<code>make_data1_nominal</code>	a boolean. If TRUE (default), DATA1 is recoded as a dummy-coded matrix. If FALSE, DATA1 is a dummy-coded matrix.
<code>make_data2_nominal</code>	a boolean. If TRUE (default), DATA2 is recoded as a dummy-coded matrix. If FALSE, DATA2 is a dummy-coded matrix.
DESIGN	a design matrix to indicate if rows belong to groups.
<code>make_design_nominal</code>	a boolean. If TRUE (default), DESIGN is a vector that indicates groups (and will be dummy-coded). If FALSE, DESIGN is a dummy-coded matrix.
<code>weights1</code>	a diagonal matrix or column-vector of weights for the columns of DATA1
<code>weights2</code>	a diagonal matrix or column-vector of weights for the columns of DATA2
<code>symmetric</code>	a boolean. If TRUE (default) symmetric factor scores for rows.
<code>graphs</code>	a boolean. If TRUE (default), graphs and plots are provided (via tepGraphs)
<code>k</code>	number of components to return.

Details

This implementation of Partial Least Squares is for two categorical data sets (Beaton et al., 2013), and based on the PLS method proposed by Tucker (1958) and again by Bookstein (1994).

Value

See [epCA](#) (and also [coreCA](#)) for details on what is returned. In addition to the values returned:

W1	Weights for columns of DATA1, replaces M from <code>coreCA</code> .
W2	Weights for columns of DATA2, replaces W from <code>coreCA</code> .
lx	latent variables from DATA1 computed for observations
ly	latent variables from DATA2 computed for observations

Author(s)

Derek Beaton, Hervé Abdi

References

- Tucker, L. R. (1958). An inter-battery method of factor analysis. *Psychometrika*, 23(2), 111–136.
- Bookstein, F., (1994). Partial least squares: a dose–response model for measurement in the behavioral and brain sciences. *Psychology* 5 (23)
- Abdi, H. (2007). Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD). In N.J. Salkind (Ed.): *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): Sage. pp. 907-912.
- Krishnan, A., Williams, L. J., McIntosh, A. R., & Abdi, H. (2011). Partial Least Squares (PLS) methods for neuroimaging: A tutorial and review. *NeuroImage*, 56(2), 455 – 475.
- Beaton, D., Filbey, F., & Abdi H. (in press, 2013). Integrating partial least squares correlation and correspondence analysis for nominal data. In Abdi, H., Chin, W., Esposito Vinzi, V., Russolillo, G., & Trinchera, L. (Eds.), *New Perspectives in Partial Least Squares and Related Methods*. New York: Springer Verlag.

See Also

[coreCA](#), [epCA](#), [epMCA](#), [tepDICA](#)

Examples

```
data(snps.druguse)
plsca.res <- tepPLSCA(snps.druguse$DATA1, snps.druguse$DATA2,
  make_data1_nominal=TRUE, make_data2_nominal=TRUE)
```

texpoDesignCheck	<i>texpoDesignCheck</i>
------------------	-------------------------

Description

TExPosition’s DESIGN matrix check function. Calls into ExPosition’s [designCheck](#).

Usage

```
texpoDesignCheck(DATA = NULL, DESIGN = NULL, make_design_nominal = TRUE, force_bary=FALSE)
```

Arguments

DATA	original data that should be matched to a design matrix
DESIGN	a column vector with levels for observations or a dummy-coded matrix
make_design_nominal	a boolean. Will make DESIGN nominal if TRUE (default).
force_bary	a boolean. If TRUE, it forces the check for barycentric methods (tepDICA, tepBADA). If FALSE, designCheck is performed.

Details

For BADA & DICA, execution stops if:

1. DESIGN has more columns (groups) than observations,
2. DESIGN has only 1 column (group),
- or 3. DESIGN has at least 1 occurrence where an observation is the only observation in a group (i.e., `colSums(DESIGN)==1` at least once).

Value

DESIGN dummy-coded design matrix

Author(s)

Derek Beaton

Index

- * **graphs**
 - tepGraphs, 15
- * **misc**
 - calculateLVConstraints, 3
 - fastEucCalc, 4
 - fii2fi, 4
 - R2, 9
 - tepGraphs, 15
 - texpoDesignCheck, 20
- * **multivariate**
 - fii2fi, 4
 - R2, 9
 - tepBADA, 10
 - tepDICA, 11
 - tepGPLS, 13
 - tepGraphs, 15
 - tepPLS, 17
 - tepPLSCA, 18
 - TExPosition-package, 2
- * **package**
 - TExPosition-package, 2
- * **print**
 - print.tepAssign, 5
 - print.tepBADA, 6
 - print.tepDICA, 6
 - print.tepGPLS, 7
 - print.tepGraphs, 7
 - print.tepPLS, 8
 - print.tepPLSCA, 8
 - print.texpoOutput, 9
- calculateLVConstraints, 3
- coreCA, 12, 13, 19, 20
- corePCA, 10, 11, 14, 15, 18
- createColorVectorsByDesign, 16
- designCheck, 20
- epCA, 12, 13, 19, 20
- epGPCA, 10, 11, 13–15, 18
- epMCA, 13, 20
- epMDS, 11
- epPCA, 11, 13, 15, 18
- expo.scale, 10, 14, 17
- ExPosition, 2
- fastEucCalc, 4
- fii2fi, 4, 10, 12
- prettyGraphs, 15–17
- prettyPlot, 16
- print.tepAssign, 5
- print.tepBADA, 6
- print.tepDICA, 6
- print.tepGPLS, 7
- print.tepGraphs, 7
- print.tepPLS, 8
- print.tepPLSCA, 8
- print.texpoOutput, 9
- R2, 9, 10, 12
- svd, 2
- tepBADA, 3, 10, 15, 18
- tepDICA, 3, 11, 15, 20
- tepGPLS, 3, 10, 13, 18
- tepGraphs, 10, 12, 14, 15, 17, 19
- tepPLS, 3, 10, 13, 15, 17
- tepPLSCA, 3, 12, 15, 18, 18
- texpoDesignCheck, 20
- TExPosition, 16
- TExPosition (TExPosition-package), 2
- TExPosition-package, 2