Package ‘TExPosition’

April 15, 2025

Type Package

Title Two-Table ExPosition

Version 2.9.0

Date 2025-03-30

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

License GPL-2

Encoding UTF-8

Depends prettyGraphs (>= 2.2.0), ExPosition (>= 2.11.0)

BugReports https://github.com/derekbeaton/ExPosition1/issues

RoxygenNote 7.3.2

NeedsCompilation no

Author Derek Beaton [aut, cre],
Jenny Rieck [aut],
Cherise R. Chin Fatt [aut],
Ju-Chi Yu [ctb],
Luke Moraglia [ctb],
Herve Abdi [aut]

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

Repository CRAN

Date/Publication 2025-04-15 04:20:02 UTC

Contents

`TExPosition-package` *TExPosition: Two-table analyses 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`.

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, Ju-Chi Yu, Luke Moraglia, 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, tepDICA, tepPLSCA

---

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(DESIGN, 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

print.tepBADA — *Print tepBADA results*

Description

Print tepBADA results.

Usage

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

Arguments

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

Author(s)

Derek Beaton, Cherise Chin-Fatt

`print.tepDICA` *Print tepDICA results*

Description

Print tepDICA results.

Usage

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

Arguments

`x`	an list that contains items to make into the tepDICA 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,
  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.
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) 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 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., Gobbini, 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`, `epMDS`

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,
  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.
- `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`

Examples

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

---

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,
    graphs = TRUE
  )
```

Arguments

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

`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:

`$fii.col` the colors that are associated to the individuals (row items; $fii).

`$fii.pch` the pch values associated to the individuals (row items; $fii).

`$fi.col` the colors that are associated to the groups ($fi).

`$fi.pch` the pch values associated to the groups ($fi).

`$fj.col` the colors that are associated to the column items ($fj).

`$fj.pch` the pch values associated to the column items ($fj).

`$constraints` 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,graphs=FALSE)
#there are only 2 components, not 3.
bada.iris.plotting.data <- tepGraphs(bada.iris,x_axis=1,y_axis=2)
```

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 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. *Psycoloquy 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, tepBADA, 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,
  symmetric = TRUE,
  graphs = TRUE,
  k = 0
)
```

Arguments

DATA1	Data matrix 1 (X), must be categorical (like MCA) or in disjunctive code see make_data1_nominal.
DATA2	Data matrix 2 (Y), must be categorical (like MCA) or in disjunctive code see make_data2_nominal.
make_data1_nominal	a boolean. If TRUE (default), DATA1 is recoded as a dummy-coded matrix. If FALSE, DATA1 is a dummy-coded matrix.
make_data2_nominal	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.
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.
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

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 coreCA.
W2	Weights for columns of DATA2, replaces W from coreCA.
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. *Psycoloquy 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` *texpoDesignCheck*

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 occurence 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