Package ‘GPAbin’

May 7, 2026

Title Unifying Multiple Biplot Visualisations into a Single Display

Version 1.1.1

Description Aligning multiple visualisations by utilising generalised orthogonal Procrustes analysis (GPA) before combining coordinates into a single biplot display as described in Nienkemper-Swanepoel, le Roux and Lubbe (2023)<doi:10.1080/03610918.2021.1914089>. This is mainly suitable to combine visualisations constructed from multiple imputations, however, it can be generalised to combine variations of visualisations from the same datasets (i.e. resamples).

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports ca, jomo, mi, mice, missMDA, mitools, stringr

Suggests testthat, knitr

Config/Needs/website rmarkdown

BugReports `https://github.com/jnienk/GPAbin/issues`

URL `https://jnienk.github.io/GPAbin/`

NeedsCompilation no

Author Johané Nienkemper-Swanepoel [aut, cre, cph] (ORCID: <`https://orcid.org/0000-0001-6086-8272`>)

Maintainer Johané Nienkemper-Swanepoel <`nienkemperj@sun.ac.za`>

Repository CRAN

Date/Publication 2025-10-02 16:50:08 UTC

Contents

`biplFig` *Biplot function*

Description

Creates a multiple correspondence analysis (MCA) biplot

Usage

```
biplFig(
  missbp,
  Z.col = "#61223b",
  CLP.col = "#b79962",
  Z.pch = 19,
  CLP.pch = 15,
  Z.cex = 1.5,
  CLP.cex = 1.7,
  title = ""
)
```

Arguments

`missbp`	An object of class `missbp` obtained from preceding function `missmi()`
`Z.col`	Colour of sample coordinates
`CLP.col`	Colour of category level point coordinates
`Z.pch`	Plotting character of sample coordinates
`CLP.pch`	Plotting character of category level point coordinates
`Z.cex`	Size of plotting character for sample points
`CLP.cex`	Size of plotting character for category level point points
`title`	Title of the plot

Value

- If `compdat` = `NULL` in `evalMeas`, only a GPAbin biplot will be constructed.
- If a complete data set (`compdat`) was specified in `evalMeas`, two biplots will be constructed: (1) Complete MCA biplot and (2) GPAbin biplot.

Examples

```
data(implist)
missbp <- missmi(implist)|> DRT() |> GPAbin() |> biplFig()
```

`CLPpred` *Category level prediction*

Description

Predicts category levels from an MCA based biplot using the distances between coordinates

Usage

```
CLPpred(CLPs = CLPs, Zs = Zs, p = p, n = n, lvls = lvls, datIN = datIN)
```

Arguments

`CLPs`	Category level point coordinates
`Zs`	Sample coordinates
`p`	Number of variables
`n`	Number of samples
`lvls`	Names of category levels
`datIN`	Input data from which `CLPs` and `Zs` are obtained

Value

`predCL`	Final predicted categorical data set

`compdat` *Complete data example*

Description

Simulated data example

Format

A data frame with 1000 rows and 5 columns.

Details

V1 Variable 1

V1 Variable 2

V1 Variable 3

V1 Variable 4

V1 Variable 5

Source

Simulated data from a uniform distribution that is categorised into levels.

`DRT` *Dimension reduction*

Description

Multiple correspondence analysis is performed on the multiple imputed datasets

Usage

```
DRT(missbp, method = c("MCA"))
```

Arguments

`missbp`	An object of class `missbp` obtained from preceding function `missmi()`
`method`	Select a dimension reduction technique. In the current version `MCA` is available.

Value

The `missbp` object is appended with the following objects:

`Z`	List of sample coordinates
`CLP`	List of category level point coordinates
`lvls`	List of category level names
`m`	Number of multiple imputations

See also `missmi` and `impute`.

Examples

```
data(implist)
missbp <- missmi(implist) |> DRT()
```

`evalMeas` *Evaluation measures when complete data is available*

Description

Calculates measures of comparison based on distances between two configurations in two dimensions.

Usage

```
evalMeas(missbp, compdat = NULL)
```

Arguments

`missbp`	An object of class `missbp` obtained from preceding function `missmi()`.
`compdat`	Complete data matrix representing the input data of `missmi()`

Value

The `missbp` object is appended with the following objects:

`eval`	Returns a data table with five evaluation measures: Procrustes Statistic (PS), Similarity Proportion (SP), Response Profile Recovery (RPR), Absolute Mean Bias (AMB), Root Mean Squared Bias (RMSB)
`GPApred`	A dataframe representing predicted categorical responses from the GPAbin biplot.
`compPred`	A dataframe representing predicted categorical responses from the complete MCA biplot.
`compZs`	Sample coordinates for the MCA biplot of the complete data.
`compCLPs`	CLPs for the MCA biplot of the complete data.
`complvls`	Names of the CLPs for the MCA biplot of the complete data.

See also `missmi`, `impute`, `DRT` and `GPAbin`.

For more detail, refer to Nienkemper-Swanepoel, J., le Roux, N. J., & Gardner-Lubbe, S. (2021). GPAbin: unifying visualizations of multiple imputations for missing values. Communications in Statistics - Simulation and Computation, 52(6), 2666–2685. https://doi.org/10.1080/03610918.2021.1914089.

Examples

```
data(compdat)
data(implist)
missbp <- missmi(implist) |> DRT() |> GPAbin() |> evalMeas(compdat=compdat)
```

`GPA` *Generalised Orthogonal Procrustes Analysis*

Description

This function contains the OPA function to compare two configurations and the GPA function for multiple configuration comparisons

Usage

```
GPA(Xk, G.target = NULL, iter = 500, eps = 0.001)
```

Arguments

`Xk`	list containing the testee configurations which is updated on #each iteration
`G.target`	Target configuration. If not specified the centroid configuration will be used as the target
`iter`	Number of iterations allowed before convergence
`eps`	Threshold value for convergence of the alogrithm

Value

`Xk.F`	List containing the updated testee configurations
`sk.F`	Vector containing the final scaling factors
`Qk.F`	List containing the final rotation matrices
`Gmat`	Final target configuration
`sum.sq`	Final minimised sum of squared distance

`GPAbin` *Function to unify coordinates of multiple configurations*

Description

Combines multiple configurations from dimension reduction solutions applied to multiple imputed data sets

Usage

```
GPAbin(missbp, G.target = NULL)
```

Arguments

`missbp`	An object of class `missmi` obtained from preceding function `missmi()`
`G.target`	Target configuration. If not specified the centroid configuration will be used as the target.

Value

The `missbp` object is appended with the following objects:

`Z.GPA.list`	List containing the sample coordinates for each MI after GPA
`CLP.GPA.list`	List containing the CLPs for each MI after GPA
`G.target`	Target configuration
`Z.GPAbin`	Sample coordinates for the GPAbin biplot
`CLP.GPAbin`	CLPs for the GPAbin biplot

See also `missmi`, `impute` and `DRT`.

For more detail, refer to Nienkemper-Swanepoel, J., le Roux, N. J., & Gardner-Lubbe, S. (2021). GPAbin: unifying visualizations of multiple imputations for missing values. Communications in Statistics - Simulation and Computation, 52(6), 2666–2685. https://doi.org/10.1080/03610918.2021.1914089.

Examples

```
data(implist)
missbp <- missmi(implist) |> DRT() |> GPAbin()
```

`implist` *List of multiple imputed data sets*

Description

Five multiple imputations of `missdat`

Format

List containing five multiple imputations of `missdat`. Each list item a data frame with 1000 rows and 5 columns.

Details

V1 Variable 1

V1 Variable 2

V1 Variable 3

V1 Variable 4

V1 Variable 5

Source

simulated example data imputed with mice::mice(missdat, m=5, method="polyreg", maxit=10, remove.collinear=FALSE, printFlag = FALSE)

`impute` *Multiple imputation*

Description

Choose between four available multiple imputation strategies in `R`.

Usage

```
impute(missbp, imp.method = c("MIMCA", "jomo", "DPMPM", "mice"), m = 5)
```

Arguments

`missbp`	An object of class `missmi` obtained from preceding function `missmi()`.
`imp.method`	Select one of four imputation methods: `MIMCA`, `jomo`, `DPMPM`, `mice`
`m`	Number of multiple imputations

Value

The `missbp` object is appended with the following object:

`dataimp`	List of imputed data

See also `MIMCA`, `jomo1cat` and `mi` and `mice`.

Examples

```
data(missdat)
missbp <- missmi(missdat) |> impute(imp.method="DPMPM", m=5)
```

`missdat` *Missing data example*

Description

`compdat` containing approximately 35% simulated missing values according to a missing at random (MAR) missing data mechanism

Format

A data frame with 1000 rows and 5 columns.

Details

V1 Variable 1

V1 Variable 2

V1 Variable 3

V1 Variable 4

V1 Variable 5

Source

Simulated data from a uniform distribution that is categorised into levels.

`missmi` *First step before constructing unified biplots*

Description

This function produces a list of elements to be used when producing a GPAbin biplot.

Usage

```
missmi(data)
```

Arguments

`data`	input data frame or list

Value

`X`	The processed data
`m`	Number of multiple imputations applied
`n`	The number of samples
`p`	The number of variables
`miss_pct`	Percentage of missing values

Examples

```
data(missdat)
missbp <- missmi(missdat)
data(implist)
missbp <- missmi(implist)
```

`OPA` *Orthogonal Procrustes Analysis*

Description

This function performs Orthogonal Procrustes Analysis on centred data

Usage

```
OPA(missbp, compdat, centring = TRUE, dim = "2D")
```

Arguments

`missbp`	An object of class `missmi` obtained from preceding function `missmi()`
`compdat`	Complete data set, only available for simulated data examples.
`centring`	Logical argument to apply centering, default is `TRUE`.
`dim`	Number of dimensions to use in final solutions (`2D` or `All` available dimensions.)

Value

`ProcStat`	Procrustes Statistic
`compZ`	Sample coordinates representing the complete data set
`compCLP`	Category level point coordinates representing the complete data set
`complvls`	Category levels
`compdat`	Complete data set, only available for simulated data examples

---

`print.missmi` *Generic print function for objects of class missmi*

Description

This function is used to print output when the missmi biplot object is created.

Usage

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

Arguments

`x`	an object of class `missmi`.
`...`	additional arguments.

Value

This function will not produce a return value, it is called for side effects.

Examples

```
data(missdat)
missbp <- missmi(missdat)
data(implist)
missbp <- missmi(implist)
print(missbp)
```

Index