Package 'thectar'

November 15, 2019

Type Package

Title Hermeneutic Content Analysis

Version 1.0.0

Date 2019-11-06

Author Lena Berger

Maintainer Lena Berger <bergerl@email.arizona.edu>

Description A collection of tools for performing the second step of
Hermeneutic Content Analysis (HCA; see Bergman, 2010,
<doi:10.4135/9781506335193>). The current version
provides tools for implementing multidimensional scaling (MDS) within
an HCA framework. The tools offer assistance in data preparation, modeling,
and interpretation of MDS maps. The data preparation tools help
users to create similarity matrices based on co-occurrence or sorting
data. The modelling tools allow users to optimize the modelling process
by systematically calculating Stress values for a set of p points out of n
points. The tools assisting interpretation allow users to highlight the
highest similarities either per point or overall in an MDS representation.

Imports smacof, graphics, utils, proxy

Depends R (>= 2.10)

License GPL-3

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-11-15 19:40:02 UTC

R topics documented:

1

---

highlall   *Highlight highest similarities overall (highlall)*

Description

highlall draws the highest similarities (overall) into an MDS graph.

Usage

```
highlall(similarity, results, quantile = 30, col = "black",
  coordinates = FALSE, add = FALSE, xlim = NULL, cex.labels = 0.8,
  main = NULL)
```

Arguments

similarity	Similarity matrix.
results	Results of smacofSym or a set of coordinates.
quantile	Percentage of the highest similarities that should be drawn. Default is 30.
col	Color of the graph. Default is black.
coordinates	If TRUE, the input results consist of a set of coordinates. Default is FALSE.
add	If TRUE, the points will be added to the existing plot. Default is FALSE.
xlim	Numeric vector giving the x coordinates range.
cex.labels	Numeric value indicating font size of the labels.
main	Title of the plot.

Details

This function is applicable to an MDS solution computed with the package 'smacof' (Mair, De Leeuw, Borg, & Groenen) or a set of coordinates. It adds the quantile percent of highest similarities, as indicated by the similarity matrix similarity, to the plot of the respective map. The objects must occur in the same order in the results/ coodinates and the similarity matrix.

Value

Nothing.

See Also

smacofSym.

Examples

```
## Calculating an MDS using the package 'smacof' and showing the 25%
## highest similarities
data(SDG_coocurrence)
SDG_coocurrence <- SDG_coocurrence[,-2] # Drop second column
similarities <- simi(SDG_coocurrence, method = "as")
dissimilarities <- 1 - similarities
res <- smacof::smacofSym(dissimilarities, type = "ordinal")
highlall(similarities, res, quantile = 25, main = "25% Highest Similarities")
```

---

highlpoints   *Highlight highest similarities per point (highlpoints)*

Description

highlpoints draws the highest similarities (per point) into an MDS graph.

Usage

```
highlpoints(similarity, results, links = 3, col = "black",
  coordinates = FALSE, add = FALSE, xlim = NULL, cex.labels = 0.8,
  main = NULL)
```

Arguments

similarity	Similarity matrix.
results	Results of smacofSym or a set of coordinates.
links	Number of similarities that should be drawn per point. Default is 3.
col	Color of the points in the graph. Default is black.
coordinates	If TRUE, the input results consist of a set of coordinates. Default is FALSE.
add	If TRUE, the points will be added to the existing plot. Default is FALSE.
xlim	Numeric vector giving the x coordinates range.
cex.labels	Numeric value indicating font size of the labels.
main	Title of the plot.

Details

This function is applicable to an MDS solution computed with the package 'smacof' (Mair, De Leeuw, Borg, & Groenen) or a set of coordinates. It adds the link highest similarities per point, as indicated by the similarity matrix similarity, to the plot of the respective map. The links belonging to one point are displayed in the same color. If there is more than one similarity on the last rank link, all will be shown. The objects must occur in the same order in the results/ coodinates and the similarity matrix.

Value

Nothing.

See Also

[smacofSym](#).

Examples

```
## Calculating an MDS using the package 'smacof' and showing the 3 highest
## similarities per point
data(SDG_coocurrence)
SDG_coocurrence <- SDG_coocurrence[,-2] # Drop second column
similarities <- simi(SDG_coocurrence, method = "as")
dissimilarities <- 1 - similarities
res <- smacof::smacofSym(dissimilarities, type = "ordinal")
highlpoints(similarities, res, links = 3,
main = "3 Highest Similarities Per Point")
```

---

lscomb   *Lowering Stress by comparing combinations (lscomb)*

Description

lscomb calculates Stress levels for all combinations of p out of n categories

Usage

```
lscomb(data, points, method = "as", single = TRUE, comments = FALSE,
  type = "ordinal", weight = TRUE)
```

Arguments

data	Dataset; the first column must be the ID of the unit of comparison, the other columns must be categories; see dataset requirements simi.
points	Number of categories that should be included in the model (p < n, wherein n equals the number of categories in the dataset).
method	Specifies the similarity index used to compute the similarity matrix, choose between "as" (Association Strength Index), "jaccard" (Jaccard Index), "cosine" (Cosine Index), and "inclusion" (Inclusion Index). Default is as.
single	If TRUE, single mentionings (i.e. one respondent mentioning just one category) are included when calculating the similarity matrix. Default is TRUE.
comments	If TRUE, comments relating to exclusion or possible exclusion of categories and respondents are displayed. Default is FALSE.
type	Specifies the type of MDS model used (for more details see the package 'smacof' of Mair, De Leeuw, Borg, & Groenen). Default is ordinal.
weight	If TRUE, the MDS model is calcualted using weights, i.e., similarities of zero are excluded. Default is TRUE.

Details

This function is applicable to co-occurrence data. It shows the resulting Stress values for all combinations of p out of n categories. The output consists in a table showing which categories have been included and the resulting Stress values. The table is sorted such that the lowest Stress level occurs at the top. The MDS model is computed using the package 'smacof' (Mair, De Leeuw, Borg, & Groenen). The first column of the input matrix `data` should contain the ID of the unit of comparison and the following columns the categories for which the similarity matrices and MDS maps are calculated (see also the description of `simi`). Note that the purpose of this function is to assist modeling by helping to identify potential problems. It is not, however, meant to be used for excluding categories based solely on measures of fit and without substantial justification.

Value

Matrix showing the categories included and the Stress values of the respective MDS models.

See Also

[smacofSym](#) for details on calculating MDS representations, [simi](#) for details on calculating similarity matrices.

Examples

```
## Calculate Stress values for 5 out of 7 SDGs
data(SDG_coocurrence)
SDG_coocurrence <- SDG_coocurrence[,-2] # Drop second column
input <- SDG_coocurrence[,1:8] # For computational reasons, we will work
                              # with 7 SDGs.
stress <- lscomb(input, points = 5)
stress
```

---

lspoints   *Lowering Stress by excluding points (lspoints)*

Description

`lspoints` calculates Stress values for all combinations with n - 1 categories.

Usage

```
lspoints(data, method = "as", single = TRUE, comments = FALSE,
  type = "ordinal", weight = TRUE)
```

Arguments

data   Dataset; the first column must be the ID of the unit of comparison, the other
                columns must be categories; see dataset requirements simi.

method	Specifies the similarity index used to compute the similarity matrix, choose between `"as"` (Association Strength Index), `"jaccard"` (Jaccard Index), `"cosine"` (Cosine Index), and `"inclusion"` (Inclusion Index). Default is `as`.
single	If TRUE, single mentionings (i.e. one respondent mentioning just one category) are included when calculating the similarity matrix.
comments	If TRUE, comments relating to exclusion or possible exclusion of categories and respondents are displayed. Default is FALSE.
type	Specifies the type of MDS model used (for more details see the package 'smacof' by Mair, De Leeuw, Borg, & Groenen). Default is `ordinal`.
weight	If TRUE, the MDS model is calcualted using weights, i.e., similarities of zero are excluded. Default is TRUE.

Details

This function is applicable to co-occurrence data. It shows the resulting Stress values when single categories are excluded. The output consists in a table showing which categories have been excluded and the resulting Stress values. The table is sorted such that the the lowest Stress level occurs at the top. The MDS model is computed using the package 'smacof' (Mair, De Leeuw, Borg, & Groenen). The first column of the input matrix `data` should contain the ID of the unit of comparison, and the following columns the categories for which the similarity matrices and MDS maps are calculated (see also the description of `simi`). Note that the purpose of this function is to assist modeling by helping to identify potential problems. It is not, however, meant to be used for excluding categories based solely on measures of fit and without substantial justification.

Value

Matrix showing the categories excluded and the Stress values of the respective MDS models.

See Also

[smacofSym](#) for details on calculating MDS representations, [simi](#) for details on calculating similarity matrices.

Examples

```
## Calculate Stress values for all combinations with n - 1 (i.e., 16) SDGs
data(SDG_coocurrence)
SDG_coocurrence <- SDG_coocurrence[,-2] # Drop second column
stress <- lspoints(SDG_coocurrence)
stress
```

| SDG_coocurrence | *Co-occurence of Sustainable Development Goals (SDGs)* |

Description

A dataset containing rating of SDGs in terms of importance by 23 respondents

Usage

```
SDG_coocurrence
```

Format

A data frame with 46 rows and 19 variables:

ID ID number of respondent

Important_q Indicates if selection represents important goals (1) or unimportant goals (0) as perceived by respondent

No_Poverty SDG 1 on no poverty

Zero_Hunger SDG 2 on zero hunger

Good_Health_and_Well_Being SDG 3 on good health and wellbeing

Quality_Education SDG 4 on quality education

Gender_Equality SDG 5 on gender equality

Clean_Water_and_Sanitation SDG 6 on clean water and sanitation

Affordable_and_Clean_Energy SDG 7 on affordable and clean energy

Decent_Work_and_Economic_Growth SDG 8 on decent work and economic growth

Industry_Innovation_and_Infrastructure SDG 9 on industry innovation and infrastructure

Reduced_Inequalities SDG 10 on reduced inequalities

Sustainable_Cities_and_Communities SDG 11 on sustainable cities and communities

Responsible_Consumption_and_Production SDG 12 on responsible consumption and production

Climate_Action SDG 13 on climate action

Life_Below_Water SDG 14 on life below water

Life_on_Land SDG 15 on life on land

Peace_Justice_and_Strong_Institutions SDG 16 on peace, justice, and strong institutions

Partnerships_for_the_Goals SDG 17 on partnerships for the goals

Source

Data collected by the Social Research and Methodology Group, University of Basel, Switzerland. For more details on the SDGs see <www.sustainabledevelopment.un.org>.

SDG_grouping   *Grouping of Sustainable Development Goals (SDGs)*

Description

A dataset containing grouping of SDGs by 23 respondents

Usage

SDG_grouping

Format

A data frame with 23 rows and 18 variables:

ID ID number of respondent

No_Poverty SDG 1 on no poverty

Zero_Hunger SDG 2 on zero hunger

Good_Health_and_Well_Being SDG 3 on good health and wellbeing

Quality_Education SDG 4 on quality education

Gender_Equality SDG 5 on gender equality

Clean_Water_and_Sanitation SDG 6 on clean water and sanitation

Affordable_and_Clean_Energy SDG 7 on affordable and clean energy

Decent_Work_and_Economic_Growth SDG 8 on decent work and economic growth

Industry_Innovation_and_Infrastructure SDG 9 on industry innovation and infrastructure

Reduced_Inequalities SDG 10 on reduced inequalities

Sustainable_Cities_and_Communities SDG 11 on sustainable cities and communities

Responsible_Consumption_and_Production SDG 12 on responsible consumption and production

Climate_Action SDG 13 on climate action

Life_Below_Water SDG 14 on life below water

Life_on_Land SDG 15 on life on land

Peace_Justice_and_Strong_Institutions SDG 16 on peace, justice, and strong institutions

Partnerships_for_the_Goals SDG 17 on partnerships for the goals

Source

Data collected by the Social Research and Methodology Group, University of Basel, Switzerland. For more details on the SDGs see <www.sustainabledevelopment.un.org>.

---

simi *Similarity matrix (simi)*

Description

simi calculates a similarity matrix for co-occurrence data.

Usage

```
simi(data, method = c("sort", "aggregate", "dichotomize", "as",
  "jaccard", "cosine", "inclusion"), single = TRUE, comments = TRUE)
```

Arguments

data	Dataset; the first column must be the ID of the unit of comparison and all other columns must be categories.
method	Specifies the output, choose between "sort" (sorted version of the data), "aggregate" (aggregated version of the data), "dichotomize" (dichotomized version of the data), "as" (similarity matrix using Association Strength Index), "jaccard" (similarity matrix using Jaccard Index), "cosine" (similarity matrix using Cosine Index), and "inclusion" (similarity matrix using Inclusion Index). Default is sort.
single	If TRUE, single mentionings (i.e. one respondent mentioning just one category) are included. Default is TRUE.
comments	If TRUE, comments relating to exclusion or possible exclusion of categories and respondents are displayed. Default is TRUE.

Details

This function applies to co-occurrence data. It calculates a similarity matrix using one of the following indices: Association Strength, Jaccard, Cosine, or Inclusion (for a detailed discussion see van Eck & Waltman, 2009, <doi:10.1002/asi.21075>). Additionally, the function can also generate a sorted, aggregated, or dichotomized version of the input data table. The first column of the input matrix should contain the ID of the unit of comparison, and the following columns the categories for which the similarity is calculated. Lines belonging to the same unit of comparison (i.e. same ID) will be combined. simi is particularly suitable for not sorted, not aggregated, or not dichotomized datasets. For datasets already sorted, aggregated, and dichotomized, the package proxy of Meyer and Buchta offers an alternative to calculate similarity matrices. simi does not work with missing data.

Value

Sorted, aggregated, or dichotomized dataset, or similarity matrix.

See Also

[dist](#) from the package 'proxy for alternative ways to calculate similarity matrices; van Eck and Waltman (2009, <doi:10.1002/asi.21075>) for a detailed discussion on similaritiy measues.

Examples

```
## Calculate similarities using a dichotomized dataset
data(SDG_coocurrence)
SDG_coocurrence <- SDG_coocurrence[,-2] # Drop second column
similarity <- simi(SDG_coocurrence, method = "as", comments = FALSE)
head(similarity)
```

---

simicount   *Similarity matrix by counting (simicount)*

Description

simicount calculates a similarity matrix for sorting data.

Usage

```
simicount(data)
```

Arguments

data   Dataset; one row represents one sorting, objects in one pile must have the same
                    number.

Details

This function is applicable to sorting data. It creates a similarity matrix showing how often two
objects were in the same pile. Each line of the dataset should refer to one sorting. The first column
of the input matrix should contain the ID of the sorting; the following columns refer to the objects
that have been sorted. The allocation of objects to piles is indicated with numbers; for each line, the
objects that were sorted into the same pile are given the same number (e.g. all objects with a "1"
are in one pile, all objects with a "2" are in one pile, etc.). This function does not work with missing
values.

Value

Similarity matrix.

Examples

```
## Calculating similarities using sorted data
data(SDG_grouping)
similarities <- simicount(SDG_grouping)
head(similarities)
```

Index