Package 'ResIN'

October 4, 2024

Type Package

Title Response Item Networks

Version 2.0.0

Maintainer Philip Warncke <pwarncke@live.unc.edu>

Description
Contains various tools to perform and visualize Response Item Networks ('ResIN's'). 'ResIN' binarizes ordered-categorical and qualitative response choices from (survey) data, calculates pairwise associations and maps the location of each item response as a node in a force-directed network. Please refer to <https://www.resinmethod.net/> for more details.

License GPL-3

URL https://github.com/pwarncke77/ResIN

BugReports https://github.com/pwarncke77/ResIN/issues

Depends R (>= 4.2.0)

Imports psych, ggplot2 (>= 3.4.4), dplyr (>= 1.1.3), fastDummies, qgraph, igraph, wCorr, Matrix, DirectedClustering, foreach, parallelly, parallel, doSNOW, readr, shadowtext (>= 0.1.4)

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Philip Warncke [cre, aut],
Dino Carpentras [aut],
Adrian Lüders [aut]

Repository CRAN

Date/Publication 2024-10-04 10:40:03 UTC

Contents

`Bootstrap_example` *Output example for a bootstrapping analysis conducted with the ResIN package.*

Description

Output example for a bootstrapping analysis conducted with the ResIN package.

Usage

```
data(Bootstrap_example)
```

Format

An object of class `"RDS"`

References

This dataset was made available by Lüders et.al. 2024.

Examples

```
data(Bootstrap_example)
str(Bootstrap_example)
```

BrJSocPsychol_2024 *Source data for Lüders, A., Carpentras, D. and Quayle, M., 2024. Attitude networks as intergroup realities: Using network-modelling to research attitude-identity relationships in polarized political contexts. British Journal of Social Psychology, 63(1), pp.37-51.*

Description

The sample of N = 402 paid participants through the crowd working platform Prolific Academic. The core of the survey consists of A set of eight political attitude items abortion, immigration, gun control, and gay marriage. Each item followed a 5-point scale ranging from strong disagreement to strong agreement. The survey also includes items on partisanship, affective polarization, and a short vignette experiment.

Usage

```
data(BrJSocPsychol_2024)
```

Format

An object of class `"data.frame"`

References

This dataset was made available by Lüders et.al. 2024.

Examples

```
data(BrJSocPsychol_2024)
head(BrJSocPsychol_2024)
```

lik_data *Likert-type simulated data for "ResIN" package examples*

Description

An artificially created data-set (n=1000) of 12, 5-point Likert data. Modeled on the basis of a standard normal data-generating process. Likert scales contain 20 percent uncorrelated, homoscedastic measurement error. This data-set is used for the examples in the "ResIN" package vignette.

Usage

```
data(lik_data)
```

Format

An object of class "data.frame"

References

This data set was artificially created for the ResIN package.

Examples

```
data(lik_data)
head(lik_data)
```

ResIN *ResIN*

Description

Performs Response Item-Network (ResIN) analysis

Usage

```
ResIN(
  df,
  node_vars = NULL,
  left_anchor = NULL,
  cor_method = "auto",
  weights = NULL,
  method_wCorr = "Polychoric",
  poly_ncor = 2,
  neg_offset = 0,
  ResIN_scores = TRUE,
  remove_negative = TRUE,
  EBICglasso = FALSE,
  EBICglasso_arglist = NULL,
  remove_nonsignificant = FALSE,
  sign_threshold = 0.05,
  node_covars = NULL,
  node_costats = NULL,
  network_stats = TRUE,
  detect_clusters = FALSE,
  cluster_method = NULL,
  cluster_arglist = NULL,
  cluster_assignment = TRUE,
  seed = NULL,
  generate_ggplot = TRUE,
```

```
  plot_ggplot = TRUE,
  plot_whichstat = NULL,
  plot_edgestat = NULL,
  color_palette = "RdBu",
  plot_responselabels = TRUE,
  response_levels = NULL,
  plot_title = NULL,
  save_input = TRUE
)
```

Arguments

df	A data-frame object containing the raw data.
node_vars	An optional character vector detailing the attitude item columns to be selected for ResIN analysis (i.e. the subset of attitude variables in df).
left_anchor	An optional character scalar indicating a particular response node which determines the spatial orientation of the ResIN latent space. If this response node does not appear on the left-hand side, the x-plane will be inverted. This ensures consistent interpretation of the latent space across multiple iterations (e.g. in bootstrapping analysis). Defaults to NULL (no adjustment to orientation is taken.)
cor_method	Which correlation method should be used? Defaults to "auto" which applies the cor_auto function from the qgraph package. Possible arguments are "auto", "pearson", "kendall", and "spearman".
weights	An optional continuous vector of survey weights. Should have the same length as number of observations in df. If weights are provided, weighted correlation matrix will be estimated with the weightedCorr function from the wCorr package.
method_wCorr	If weights are supplied, which method for weighted correlations should be used? Defaults to "Polychoric". See wCorr::weightedCorr for all correlation options.
poly_ncor	How many CPU cores should be used to estimate polychoric correlation matrix? Only used if cor_method = "polychoric".
neg_offset	Should negative correlations be offset to avoid small correlation pairs disappearing? Defaults to 0. Any positive number between 0 and 1 may be supplied instead.
ResIN_scores	Should spatial scores be calculated for every individual. Defaults to TRUE. Function obtains the mean positional score on the major (x-axis) and minor (y-axis). Further versions of this package will include more sophisticated scoring techniques.
remove_negative	Should all negative correlations be removed? Defaults to TRUE (highly recommended). Setting to FALSE makes it impossible to estimate a force-directed network layout. Function will use igraph::layout_nicely instead.

`EBICglasso` Should a sparse, Gaussian-LASSO ResIN network be estimated? Defaults to FALSE. If set to TRUE, `EBICglasso` function from the `qgraph` packages performs regularization on (nearest positive-semi-definite) ResIN correlation matrix.

`EBICglasso_arglist`
An argument list feeding additional instructions to the `EBICglasso` function if `EBICglasso` is set to TRUE.

`remove_nonsignificant`
Optionally, should non-significant edges be removed from the ResIN network? Defaults to FALSE. Note that this option is incompatible with EBICglasso and weighted correlations.

`sign_threshold` At what p-value threshold should non-significant edges be removed? Defaults to 0.05.

`node_covars` An optional character string selecting quantitative covariates that can be used to enhance ResIN analysis. Typically, these covariates provide grouped summary statistics for item response nodes. (E.g.: What is the average age or income level of respondents who selected a particular item response?) Variable names specified here should match existing columns in `df`.

`node_costats` If any `node_covars` are selected, what summary statistics should be estimated from them? Argument should be a character vector and call a base-R function. (E.g. `"mean"`, `"median"`, `"sd"`). Each element specified in `node_costats` is applied to each element in `node_covars` and the out-put is stored as a node-level summary statistic in the `ResIN_nodeframe`. The extra columns in `ResIN_nodeframe` are labeled according to the following template: "covariate name"_"statistic". So for the respondents mean age, the corresponding column in `ResIN_nodeframe` would be labeled as "age_mean".

`network_stats` Should common node- and graph level network statistics be extracted? Calls `qgraph::centrality_auto` and `DirectedClustering::ClustF` to the ResIN graph object to extract node-level betweenness, closeness, strength centrality, as well as the mean and standard deviation of these scores at the network level. Also estimates network expected influence, average path length, and global clustering coefficients. Defaults to TRUE. Set to FALSE if estimation takes a long time.

`detect_clusters`
Optional, should community detection be performed on item response network? Defaults to FALSE. If set to TRUE, performs a clustering method from the [igraph](https://igraph.org/r/doc/cluster_leading_eigen.html) library and stores the results in the `ResIN_nodeframe` output.

`cluster_method` A character scalar specifying the [igraph-based](https://igraph.org/r/doc/communities.html) community detection function.

`cluster_arglist`
An optional list specifying additional arguments to the selected [igraph](https://igraph.org/r/doc/communi clustering method.

`cluster_assignment`
Should individual (survey) respondents be assigned to different clusters? If set to TRUE, function will generate an n*c matrix of probabilities for each respondent to be assigned to one of c clusters. Furthermore, a vector of length n is

generated displaying the most likely cluster respondents belong to. In case of a tie between one or more clusters, a very small amount of random noise determines assignment. Both matrix and vectors are added to the `aux_objects` list. Defaults to FALSE and will be ignored if `detect_clusters` is set to FALSE.

`seed` Random seed for force-directed algorithm. Defaults to NULL (no seed is set.) If scalar integer is supplied, that seed will be set prior to analysis.

`generate_ggplot`
Should a ggplot-based visualization of the ResIN network be generated? Defaults to TRUE.

`plot_ggplot` Should a basic ggplot of the ResIN network be plotted? Defaults to TRUE. If set to FALSE, the ggplot object will not be directly returned to the console. (However, if generate_ggplot=TRUE, the plot will still be generated and stored alongside the other output objects.)

`plot_whichstat` Should a particular node-level metric be color-visualized in the ggplot output? For node cluster, specify "cluster". For the same Likert response choices or options, specify "choices". For a particular node-level co-variate please specify the name of the particular element in `node_covars` followed by a "_" and the specific `node_costats` you would like to visualize. For instance if you want the visualize average age at the node-level, you should specify "age_mean". To colorize by node centrality statistics, possible choices are "Strength", "Betweenness", "Closeness", and "ExpectedInfluence". Defaults to NULL. Make sure to supply appropriate choices to `node_covars`, `node_costats`, `detect_clusters`, and/or `network_stats` prior to setting this argument.

`plot_edgestat` Should the thickness of the edges be adjusted according to a particular co-statistic? Defaults to NULL. Possible choices are "weight" for the bi-variate correlation strength, and "edgebetweenness"

`color_palette` Optionally, you may specify the ggplot2 color palette to be applied to the plot. All options contained in [`RColorBrewer`](https://cran.r-project.org/web/packages/RColorBrewer/RColorl (for discrete colors such as cluster assignments) and [`ggplot2::scale_colour_distiller`](https://ggplo are supported. Defaults to "RdBu".

`plot_responselabels`
Should response labels be plotted via `geom_text`? Defaults to TRUE. It is recommended to set to FALSE if the network possesses a lot of nodes and/or long response choice names.

`response_levels`
An optional character vector specifying the correct order of global response levels. Only useful if all node-items follow the same convention (e.g. ranging from "strong disagreement" to "strong agreement"). The supplied vector should have the same length as the total number of response options and supply these (matching exactly) in the correct order. E.g. c("Strongly Agree", "Somewhat Agree", "Neutral", "Somewhat Disagree", "Strongly Disagree"). Defaults to NULL.

`plot_title` Optionally, a character scalar specifying the title of the ggplot output. Defaults to "ResIN plot".

`save_input` Optionally, should input data and function arguments be saved (this is necessary for running ResIN_boots_prepare function). Defaults to TRUE.

Value

An edge-list type data-frame, `ResIN_edgelist`, a node-level data-frame, `ResIN_nodeframe`, an n*2 data-frame of individual-level spatial scores along the major (x) and minor(y) axis, `ResIN_scores` a list of graph-level statistics `graph_stats` including (`graph_structuration`) and centralization (`graph_centralization`), as well as a list of auxiliary objects, `aux_objects`, including the ResIN adjacency matrix (`adj_matrix`), a numeric vector detailing which item responses belong to which item (`same_items`), and the dummy-coded item-response data-frame (`df_dummies`).

Examples

```
## Load the 12-item simulated Likert-type toy dataset
data(lik_data)

# Apply the ResIN function to toy Likert data:
ResIN_obj <- ResIN(lik_data, cor_method = "spearman", network_stats = TRUE, detect_clusters = TRUE)
```

`ResIN_boots_execute` *ResIN_boots_execute*

Description

Execute prepared ResIN bootstrap analysis

Usage

```
ResIN_boots_execute(
  ResIN_boots_prepped,
  parallel = FALSE,
  detect_cores = TRUE,
  core_offset = 0L,
  n_cores = 2L,
  inorder = FALSE
)
```

Arguments

`ResIN_boots_prepped`	A list of prepared ResIN objects for bootstrapping (outcome of the `ResIN_boots_prepare` function)
`parallel`	Should the function be executed in parallel using the `foreach` package? Defaults to FALSE. If FALSE, function will execute sequentially in a simple for loop.
`detect_cores`	Should the number of available CPU cores be automatically detected? Defaults to TRUE and is ignored when parallel is set to FALSE.

core_offset	Optionally, specify a positive integer offset that is subtracted from the number of automatically detected cores. Defaults to 0L.
n_cores	Manually specify the number of available CPU cores. Defaults to 2L and is ignored if detect_cores is set to TRUE or if parallel is set to FALSE.
inorder	Should parallel execution be done in sequential order of the ResIN_boots_prepped object?

Value

A list object containing n (bootstrapped) ResIN list objects.

Examples

```
## Load the 12-item simulated Likert-type toy dataset
data(lik_data)

# Apply the ResIN function to toy Likert data:
ResIN_obj <- ResIN(lik_data, cor_method = "spearman", network_stats = TRUE,
                      generate_ggplot = FALSE)

## Not run:
# Prepare for bootstrapping
prepped_boots <- ResIN_boots_prepare(ResIN_obj, n=5000, boots_type="permute")

# Execute the prepared bootstrap list
executed_boots <-  ResIN_boots_execute(prepped_boots, parallel = TRUE, detect_cores = TRUE)

# Extract results - here for example, the network (global)-clustering coefficient
ResIN_boots_extract(executed_boots, what = "global_clustering", summarize_results = TRUE)

## End(Not run)
```

ResIN_boots_extract *ResIN_boots_extract*

Description

Extract and summarize quantities from bootstrapped ResIN objects

Usage

```
ResIN_boots_extract(ResIN_boots_executed, what, summarize_results = FALSE)
```

Arguments

ResIN_boots_executed
: A list of prepared ResIN objects for bootstrapping (outcome of the ResIN_boots_executed function)

what
: A character vector of length one specifying the target quantity of interest. This should be a one-to-one match with the corresponding output vector (or scalar) among the bootstrapped result list (see ResIN_boots_execute). For participants' position on the x-axis of the ResIN latent space, for example, specify "scores_x".

summarize_results
: Should the extracted quantities be summarized through a series of descriptive statistics? If set to true, the minimum, maximum, mean, selected quantiles, and the standard deviation are reported. If set to FALSE (default), extracted quantities are instead returned as a list.

Value

A list object containing n (bootstrapped) ResIN list objects.

Examples

```
## Load the 12-item simulated Likert-type toy dataset
data(lik_data)

# Apply the ResIN function to toy Likert data:
ResIN_obj <- ResIN(lik_data, cor_method = "spearman", network_stats = TRUE,
                      generate_ggplot = FALSE)

## Not run:
# Prepare for bootstrapping
prepped_boots <- ResIN_boots_prepare(ResIN_obj, n=5000, boots_type="permute")

# Execute the prepared bootstrap list
executed_boots <-  ResIN_boots_execute(prepped_boots, parallel = TRUE, detect_cores = TRUE)

# Extract results - here for example, the network (global)-clustering coefficient
ResIN_boots_extract(executed_boots, what = "global_clustering", summarize_results = TRUE)

## End(Not run)
```

ResIN_boots_prepare *ResIN_boots_prepare*

Description

Prepare a ResIN-based bootstrap analysis

Usage

```
ResIN_boots_prepare(
  ResIN_object,
  n = 10000,
  boots_type = "resample",
  resample_size = NULL,
  weights = NULL,
  save_input = FALSE,
  seed = 42
)
```

Arguments

ResIN_object	A ResIN object to prepare bootstrapping workflow.
n	Bootstrapping sample size. Defaults to 10.000.
boots_type	What kind of bootstrapping should be performed? If set to "resample", function performs row-wise re-sampling of raw data (useful for e.g., sensitivity or power analysis). If set to "permute", function will randomly reshuffle raw item responses (useful e.g., for simulating null-hypothesis distributions). Defaults to "resample".
resample_size	Optional parameter determining sample size when boots_type is set to "resample". Defaults of to number of rows in raw data.
weights	An optional weights vector that can be used to adjust the re-sampling of observations. Should either be NULL (default) or a positive numeric vector of the same length as the original data.
save_input	Should all input information for each bootstrap iteration (including re-sampled/permuted data) be stored. Set to FALSE by default to save a lot of memory and disk storage.
seed	Random seed for bootstrap samples

Value

A list object containing n re-sampled or permuted copies of the raw data, along with a list of instructions for how to perform the ResIN analysis and what outputs to generate.

Examples

```
## Load the 12-item simulated Likert-type toy dataset
data(lik_data)

# Apply the ResIN function to toy Likert data:
ResIN_obj <- ResIN(lik_data, cor_method = "spearman", network_stats = TRUE,
                     generate_ggplot = FALSE)

## Not run:
# Prepare for bootstrapping
prepped_boots <- ResIN_boots_prepare(ResIN_obj, n=5000, boots_type="permute")
```

```
# Execute the prepared bootstrap list
executed_boots <- ResIN_boots_execute(prepped_boots, parallel = TRUE, detect_cores = TRUE)

# Extract results - here for example, the network (global)-clustering coefficient
ResIN_boots_extract(executed_boots, what = "global_clustering", summarize_results = TRUE)

## End(Not run)
```

ResIN_to_gephi *ResIN_to_gephi*

Description

Saves a ResIN graph as a series of csv files readable by Gephi. Source code taken from RMHogervorst / gephi

Usage

```
ResIN_to_gephi(ResIN_object, file = "ResIN_gephi.csv")
```

Arguments

ResIN_object	the output of the ResIN function (a list with class ResIN).
file	the name with .csv extension for the Gephi readable file to be output at. Defaults to "ResIN_gephi.csv".

Value

A series of csv files readable by Gephi

References

Source code was taken from: https://github.com/RMHogervorst/gephi?tab=MIT-1-ov-file#readme

Examples

```
## Not run:
## Load the 12-item simulated Likert-type ResIN toy dataset
data(lik_data)

## Run the function:
ResIN_to_gephi(ResIN(lik_data), file = "ResIN_gephi.csv")

## End(Not run)
```

ResIN_to_igraph *ResIN_to_igraph*

Description

Transforms the output of the ResIN function into an [igraph](https://igraph.org/r/doc/cluster_leading_eigen.html) object

Usage

```
ResIN_to_igraph(ResIN_object, igraph_arglist = NULL)
```

Arguments

ResIN_object	the output of the ResIN function (a list with class ResIN).
igraph_arglist	an optional argument list to be supplied to the igraph::graph_from_adjacency_matrix function. If NULL, default is: list(mode = "undirected", weighted = TRUE, diag = FALSE).

Value

A class igraph object.

References

Csardi G, Nepusz T (2006). "The igraph software package for complex network research." InterJournal, Complex Systems, 1695. https://igraph.org.

Examples

```
## Load the 12-item simulated Likert-type ResIN toy dataset
data(lik_data)

## Run the function:

igraph_output <-  ResIN_to_igraph(ResIN(lik_data))


## Plot and/or investigate as you wish:
igraph::plot.igraph(igraph_output)
```

ResIN_to_qgraph *ResIN_to_qgraph*

Description

Transforms the output of the ResIN function into an qgraph object

Usage

```
ResIN_to_qgraph(ResIN_object, qgraph_arglist = NULL)
```

Arguments

ResIN_object	the output of the ResIN function (a list with class ResIN).
qgraph_arglist	an optional argument list to be supplied to the igraph::graph_from_adjacency_matrix function. If NULL, defaults are: list(layout = "spring", maximum = 1, vsize = 6, DoNotPlot = TRUE, sampleSize = nrow(df_nodes), mar = c(3,3,3,3), normalize = FALSE)

Value

A [qgraph]https://cran.r-project.org/web/packages/qgraph/index.html graph object.

References

Epskamp S, Cramer AOJ, Waldorp LJ, Schmittmann VD, Borsboom D (2012). "qgraph: Network Visualizations of Relationships in Psychometric Data." Journal of Statistical Software, 48(4), 1–18.

Examples

```
## Load the 12-item simulated Likert-type ResIN toy dataset
data(lik_data)

## Run the function:
ResIN_qgraph <-  ResIN_to_qgraph(ResIN(lik_data))
```

Index