

Package ‘circumplex’

August 22, 2023

Type Package

Title Analysis and Visualization of Circular Data

Version 0.3.10

Description Circumplex models, which organize constructs in a circle around two underlying dimensions, are popular for studying interpersonal functioning, mood/affect, and vocational preferences/environments. This package provides tools for analyzing and visualizing circular data, including scoring functions for relevant instruments and a generalization of the bootstrapped structural summary method from Zimmermann & Wright (2017) <[doi:10.1177/1073191115621795](https://doi.org/10.1177/1073191115621795)> and functions for creating publication-ready tables and figures from the results.

License GPL-3

URL <https://github.com/jmgirard/circumplex>,
<http://circumplex.jmgirard.com/>

BugReports <https://github.com/jmgirard/circumplex/issues>

Depends R (>= 3.4)

Imports assertthat (>= 0.2.1), boot (>= 1.3-18), dplyr (>= 0.8.5),
ggforce (>= 0.3.0), ggplot2 (>= 3.3.0), htmlTable (>= 1.13.3),
magrittr (>= 1.5), purrr (>= 0.3.4), Rcpp, rlang (>= 0.4.11),
scales, stats, tibble (>= 3.0.0), tidyr (>= 1.0.0)

Suggests covr (>= 3.5.0), ggrepel, kableExtra (>= 1.1.0), knitr (>= 1.28), RColorBrewer, rmarkdown (>= 2.1), roxygen2 (>= 7.1.0),
testthat (>= 3.0.0), vdiff

LinkingTo Rcpp, RcppArmadillo (>= 0.11)

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Config/testthat/edition 3

NeedsCompilation yes

Author Jeffrey Girard [aut, cre] (<<https://orcid.org/0000-0002-7359-3746>>),
 Johannes Zimmermann [aut] (<<https://orcid.org/0000-0001-6975-2356>>),
 Aidan Wright [aut] (<<https://orcid.org/0000-0002-2369-0601>>)

Maintainer Jeffrey Girard <me@jmgirard.com>

Repository CRAN

Date/Publication 2023-08-22 07:20:05 UTC

R topics documented:

anchors	2
aw2009	3
html_render	4
instrument	4
instruments	5
ipsatize	6
items	7
jz2017	7
norms	8
octants	9
poles	10
quadrants	10
raw_iipsc	11
scales	11
score	12
ssm_analyze	13
ssm_append	15
ssm_parameters	16
ssm_plot	18
ssm_plot_circle	19
ssm_plot_contrast	20
ssm_plot_scores	21
ssm_score	22
ssm_table	23
standardize	24

Index	26
--------------	-----------

anchors

Display the anchors of a circumplex instrument

Description

Display the anchors of a circumplex instrument including the total number of anchors and each anchor's numerical value and text label. Anchors are the response options that respondents select from (e.g., 0 = No, 1 = Yes).

Usage

```
anchors(x)
```

Arguments

x Required. An object of the instrument class.

Value

The same input object. Prints text to console.

See Also

Other instrument functions: [instruments\(\)](#), [instrument\(\)](#), [items\(\)](#), [norms\(\)](#), [scales\(\)](#)

Examples

```
instrument(csip)
anchors(csip)
```

aw2009

Standardized octant scores on hypothetical circumplex scales

Description

A small example dataset containing standardized scores on eight hypothetical circumplex scales. Taken from Wright, Pincus, Conroy, & Hilsenroth (2009).

Usage

```
aw2009
```

Format

A data frame with 5 observations and 8 variables:

PA circumplex scale at 90 degrees
BC circumplex scale at 135 degrees
DE circumplex scale at 180 degrees
FG circumplex scale at 225 degrees
HI circumplex scale at 270 degrees
JK circumplex scale at 315 degrees
LM circumplex scale at 360 degrees
NO circumplex scale at 45 degrees

Source

[doi:10.1080/00223890902935696](https://doi.org/10.1080/00223890902935696)

html_render	<i>Format and render data frame as HTML table</i>
-------------	---

Description

Format a data frame as an HTML table and render it to the web viewer.

Usage

```
html_render(df, caption = NULL, align = "l", ...)
```

Arguments

df	A data frame to be rendered as an HTML table.
caption	A string to be displayed above the table.
align	A string indicating the alignment of the cells (default = "l").
...	Other arguments to pass to <code>htmlTable</code> .

Value

HTML syntax for the df table.

See Also

Other table functions: [ssm_append\(\)](#), [ssm_table\(\)](#)

instrument	<i>Load a specific instrument object</i>
------------	--

Description

The `circumplex` package includes information about numerous circumplex instruments including instructions for scoring and standardizing items to be used in conjunction with the `score` and `standardize` functions. This function loads the information for a specific instrument into memory. See the `instruments` function to list all available instruments.

Usage

```
instrument(code)
```

Arguments

code	Required. A string (e.g., "iip32") or text in non-standard evaluation (e.g., <code>iip32</code>). The code of the instrument assigned by this package and displayed in parentheses by <code>instruments()</code> .
------	---

Value

The instrument object for the requested circumplex instrument. If the function is called without a name assignment (LHS), then the object will be created in the global environment with the default name as above. Or, if a name is assigned (LHS), the object will have that name instead.

See Also

Other instrument functions: [anchors\(\)](#), [instruments\(\)](#), [items\(\)](#), [norms\(\)](#), [scales\(\)](#)

Examples

```
instrument(iip32)
instrument("iip32")
x <- instrument(iip32)
```

instruments

List all available instruments

Description

The circumplex package includes information about numerous circumplex instruments including instructions for scoring and standardizing items. Individual instruments can be loaded using the `instrument` function.

Usage

```
instruments()
```

See Also

Other instrument functions: [anchors\(\)](#), [instrument\(\)](#), [items\(\)](#), [norms\(\)](#), [scales\(\)](#)

Examples

```
instruments()
```

`ipsatize`*Ipsatize circumplex items using deviation scoring across variables*

Description

Rescore each circumplex item using deviation scoring across variables. In other words, subtract each observation's mean response from each response. This effectively removes the presence of a general factor, which can make certain circumplex fit analyses more powerful.

Usage

```
ipsatize(.data, items, na.rm = TRUE, overwrite = FALSE)
```

Arguments

<code>.data</code>	Required. A data frame containing at least circumplex scales.
<code>items</code>	Required. The variable names or column numbers for the variables in <code>.data</code> that contain circumplex items to be ipsatized.
<code>na.rm</code>	Optional. A logical that determines whether missing values should be ignored during the calculation of the mean during ipsatization (default = TRUE).
<code>overwrite</code>	Optional. A logical that determines whether the variables specified in <code>items</code> should be overwritten with ipsatized versions or alternatively preserved and new variables ending with <code>"_i"</code> should be added to the data frame (default = FALSE).

Value

A data frame that matches `.data` except that the variables specified in `items` have been rescored using ipsatization.

See Also

Other tidying functions: [score\(\)](#), [standardize\(\)](#)

Examples

```
data("raw_iipsc")
ipsatize(raw_iipsc, IIP01:IIP32)
```

`items`*Display the items of a circumplex instrument*

Description

Display the items of a circumplex instrument including the total number of items and each item's number and text. The item ordering/numbering displayed here is the same ordering/numbering assumed by the `score()` function.

Usage

```
items(x)
```

Arguments

`x` Required. An object of the instrument class.

Value

The same input object. Prints text to console.

See Also

Other instrument functions: [anchors\(\)](#), [instruments\(\)](#), [instrument\(\)](#), [norms\(\)](#), [scales\(\)](#)

Examples

```
instrument(csip)
items(csip)
```

`jz2017`*Raw octant scores on real circumplex scales with covariates*

Description

A large example dataset containing gender, raw mean scores on the Inventory of Interpersonal Problems - Short Circumplex (IIP-SC), and raw sum scores on the Personality Diagnostic Questionnaire - 4th Edition Plus (PDQ-4+).

Usage

```
jz2017
```

Format

A data frame with 1166 observations and 19 variables:

Gender Self-reported Gender
PA Domineering Problems (IIP-SC) 90 degrees
BC Vindictive Problems (IIP-SC) 135 degrees
DE Cold Problems (IIP-SC) 180 degrees
FG Socially Avoidant Problems (IIP-SC) 225 degrees
HI Nonassertive Problems(IIP-SC) 270 degrees
JK Easily Exploited Problems (IIP-SC) 315 degrees
LM Overly Nurturant Problems (IIP-SC) 360 degrees
NO Intrusive Problems (IIP-SC) 45 degrees
PARPD Paranoid PD Symptoms (PDQ-4+)
SCZPD Schizoid PD Symptoms (PDQ-4+)
SZTPD Schizotypal PD Symptoms (PDQ-4+)
ASPD Antisocial PD Symptoms (PDQ-4+)
BORPD Borderline PD Symptoms (PDQ-4+)
HISPD Histrionic PD Symptoms (PDQ-4+)
NARPD Narcissistic PD Symptoms (PDQ-4+)
AVPD Avoidant PD Symptoms (PDQ-4+)
DPNPD Dependent PD Symptoms (PDQ-4+)
OCPD Obsessive-Compulsive PD Symptoms (PDQ-4+)

Source

[doi:10.1177/1073191115621795](https://doi.org/10.1177/1073191115621795)

norms

Display the norms for a circumplex instrument

Description

Display the norms for a circumplex instrument including the total number of normative data sets available and each data set's number, sample size, population, and source reference and hyperlink. If another normative data set exists that is not yet included in the package, please let us know.

Usage

norms(x)

Arguments

x Required. An object of the instrument class.

Value

The same input object. Prints text to console.

See Also

Other instrument functions: [anchors\(\)](#), [instruments\(\)](#), [instrument\(\)](#), [items\(\)](#), [scales\(\)](#)

Examples

```
instrument(csip)
norms(csip)
```

octants

Angular displacements for octant circumplex scales

Description

Return a vector of angular displacements, in degrees, for eight equally spaced circumplex scales corresponding to the circumplex octants. Can be passed to the `angles` parameter of other functions in this package.

Usage

```
octants()
```

Value

A numeric vector with eight elements, each corresponding to the angular displacement (in degrees) of a subscale, in the following order: PA, BC, DE, FG, HI, JK, LM, NO.

Examples

```
octants()
```

poles

Angular displacements for pole circumplex scales

Description

Return a vector of angular displacements, in degrees, for four equally spaced circumplex scales corresponding to the circumplex poles. Can be passed to the `angles` parameter of other functions in this package.

Usage

```
poles()
```

Value

A numeric vector with four elements, each corresponding to the angular displacement (in degrees) of a subscale, in the following order: PA, DE, HI, LM.

Examples

```
poles()
```

quadrants

Angular displacements for quadrant circumplex scales

Description

Return a vector of angular displacements, in degrees, for four equally spaced circumplex scales corresponding to the circumplex quadrants. Can be passed to the `angles` parameter of other functions in this package.

Usage

```
quadrants()
```

Value

A numeric vector with eight elements, each corresponding to the angular displacement (in degrees) of a subscale, in the following order: BC, FG, JK, NO.

Examples

```
quadrants()
```

raw_iipsc	<i>Raw item responses on real circumplex scales</i>
-----------	---

Description

A small example dataset containing raw item responses on the Inventory of Interpersonal Problems, Short Circumplex (IIP-SC). This data set is useful for testing functions that operate on item-level data.

Usage

```
raw_iipsc
```

Format

A data frame with 10 observations and 32 variables.

scales	<i>Display the scales of a circumplex instrument</i>
--------	--

Description

Display the scales of a circumplex instrument including the total number of scales and each scale's abbreviation, hypothetical angle, and text label.

Usage

```
scales(x, items = FALSE)
```

Arguments

x	Required. An object of the instrument class.
items	Optional. A logical determining whether the items for each scale should be displayed below its other information (default = FALSE).

Value

The same input object. Prints text to console.

See Also

Other instrument functions: [anchors\(\)](#), [instruments\(\)](#), [instrument\(\)](#), [items\(\)](#), [norms\(\)](#)

Examples

```
instrument(csip)
scales(csip)
scales(csip, items = TRUE)
```

score	<i>Score circumplex scales from item responses</i>
-------	--

Description

Calculate mean scores on circumplex scales from item responses by using a set of scoring instructions, which may be loaded from the package or created as a custom data frame.

Usage

```
score(.data, items, instrument, na.rm = TRUE, prefix = "", suffix = "")
```

Arguments

<code>.data</code>	Required. A data frame containing at least circumplex scales.
<code>items</code>	Required. The variable names or column numbers for the variables in <code>.data</code> that contain all the circumplex items from a single circumplex measure, in ascending order from item 1 to item N.
<code>instrument</code>	Required. An instrument object from the package. To see the available circumplex instruments, use <code>instruments()</code> .
<code>na.rm</code>	Optional. A logical that determines if missing values should be omitted from the calculation of scores (default = TRUE). When set to TRUE, scales with missing data are essentially calculated with mean imputation.
<code>prefix</code>	Optional. A string to include at the beginning of the newly calculated scale variables' names, before <code>Abbrev</code> from <code>key</code> and <code>suffix</code> (default = "").
<code>suffix</code>	Optional. A string to include at the end of the newly calculated scale variables' names, after <code>Abbrev</code> from <code>key</code> and <code>prefix</code> (default = "").

Value

A data frame that matches `.data` except that new variables are appended that contain mean scores on each variable included in `key`.

See Also

Other tidying functions: [ipsatize\(\)](#), [standardize\(\)](#)

Examples

```
data("raw_iipsc")
instrument("iipsc")
score(raw_iipsc, IIP01:IIP32, iipsc)
```

ssm_analyze

Perform analyses using the Structural Summary Method

Description

Calculate SSM parameters with bootstrapped confidence intervals for a variety of different analysis types. Depending on what arguments are supplied, either mean-based or correlation-based analyses will be performed, one or more groups will be used to stratify the data, and contrasts between groups or measures will be calculated.

Usage

```
ssm_analyze(
  .data,
  scales,
  angles = octants(),
  measures = NULL,
  grouping = NULL,
  contrast = c("none", "test", "model"),
  boots = 2000,
  interval = 0.95,
  listwise = TRUE,
  measures_labels = NULL
)
```

Arguments

<code>.data</code>	Required. A data frame containing at least circumplex scales.
<code>scales</code>	Required. The variable names or column numbers for the variables in <code>.data</code> that contain circumplex scales to be analyzed.
<code>angles</code>	Optional. A numeric vector containing the angular displacement of each circumplex scale included in <code>scales</code> (in degrees). (default = <code>octants()</code>).
<code>measures</code>	Optional. The variable names or column numbers for one or more variables in <code>.data</code> to be correlated with the circumplex scales and analyzed using correlation-based SSM analyses. To analyze the circumplex scales using mean-based analyses, simply omit this argument or set it to <code>NULL</code> (default = <code>NULL</code>).
<code>grouping</code>	Optional. The variable name or column number for the variable in <code>.data</code> that indicates the group membership of each observation. To analyze all observations in a single group, simply omit this argument or set it to <code>NULL</code> (default = <code>NULL</code>).
<code>contrast</code>	Optional. A string indicating what type of contrast to run. Current options are "none" for no contrast, "model" to find SSM parameters for the difference scores, or "test" to find the difference between the SSM parameters. Note that only two groups or measures can be contrasted at a time (default = "none").
<code>boots</code>	Optional. A single positive integer indicating how many bootstrap resamples to use when estimating the confidence intervals (default = 2000).

interval	Optional. A single positive number between 0 and 1 (exclusive) that indicates what confidence level to use when estimating the confidence intervals (default = 0.95).
listwise	Optional. A logical indicating whether missing values should be handled by listwise deletion (TRUE) or pairwise deletion (FALSE). Note that pairwise deletion may result in different missing data patterns in each bootstrap resample and is slower to compute (default = TRUE).
measures_labels	Optional. A character vector providing a label for each measure provided in measures (in the same order) to appear in the results as well as tables and plots derived from the results. If omitted or set to NULL will default to using the measures variable names (default = NULL).

Value

A list containing the results and description of the analysis.

results	A tibble with the SSM parameter estimates
details	A list with the number of bootstrap resamples (boots), the confidence interval percentage level (interval), and the angular displacement of scales (angles)
call	A language object containing the function call that created this object
scores	A tibble containing the mean scale scores
type	A string indicating what type of SSM analysis was done

See Also

Other ssm functions: [ssm_append\(\)](#), [ssm_parameters\(\)](#), [ssm_plot\(\)](#), [ssm_score\(\)](#), [ssm_table\(\)](#)

Other analysis functions: [ssm_parameters\(\)](#), [ssm_score\(\)](#)

Examples

```
# Load example data
data("jz2017")

# Single-group mean-based SSM
ssm_analyze(jz2017, scales = PA:NO, angles = octants())

# Single-group correlation-based SSM
ssm_analyze(jz2017,
  scales = PA:NO, angles = octants(),
  measures = c(NARPD, ASPD)
)

# Multiple-group mean-based SSM
ssm_analyze(jz2017, scales = PA:NO, angles = octants(), grouping = Gender)

# Multiple-group mean-based SSM with contrast
ssm_analyze(jz2017,
  scales = PA:NO, angles = octants(), grouping = Gender,
```

```

    contrast = "model"
  )

# Single-group correlation-based SSM with contrast
ssm_analyze(jz2017,
  scales = PA:NO, angles = octants(),
  measures = c(NARPD, ASPD), contrast = "test"
)

ssm_analyze(jz2017,
  scales = PA:NO, angles = octants(), measures = c(NARPD, ASPD),
  measures_labels = c("Narcissistic", "Antisocial")
)

# Multiple-group correlation-based SSM
ssm_analyze(jz2017,
  scales = PA:NO, angles = octants(), measures = NARPD,
  grouping = Gender
)

# Multiple-group correlation-based SSM with contrast
ssm_analyze(jz2017,
  scales = PA:NO, angles = octants(), measures = NARPD,
  grouping = Gender, contrast = "test"
)

```

ssm_append

Combine SSM tables

Description

Combine SSM tables by appending them as rows.

Usage

```
ssm_append(.ssm_table, ..., caption = NULL, render = TRUE)
```

Arguments

<code>.ssm_table</code>	A data frame from the <code>ssm_table()</code> function to be the first row(s) of the combined table.
<code>...</code>	One or more additional data frames from the <code>ssm_table()</code> function to be appended to <code>.ssm_table</code> in the order of input.
<code>caption</code>	A string to be displayed above the table if rendered.
<code>render</code>	A logical indicating whether the table should be displayed in the RStudio viewer or web browser (default = TRUE).

Value

A tibble containing the information for the HTML table. As a side-effect, may also output the HTML table to the web viewer.

See Also

Other ssm functions: [ssm_analyze\(\)](#), [ssm_parameters\(\)](#), [ssm_plot\(\)](#), [ssm_score\(\)](#), [ssm_table\(\)](#)

Other table functions: [html_render\(\)](#), [ssm_table\(\)](#)

Examples

```
data("jz2017")
res1 <- ssm_analyze(jz2017, PA:NO, octants())
res2 <- ssm_analyze(jz2017, PA:NO, octants(), grouping = Gender)
tab1 <- ssm_table(res1, render = FALSE)
tab2 <- ssm_table(res2, render = FALSE)
ssm_append(tab1, tab2)
```

ssm_parameters

Calculate Structural Summary Method parameters for a set of scores

Description

Calculate SSM parameters (without confidence intervals) for a set of scores and generate a tibble with customizable labels for each parameter value. This function requires the input to be a numeric vector (or coercable to one) and returns only the parameters. See [ssm_score\(\)](#) for a similar function that calculates SSM parameters for each row of a data frame.

Usage

```
ssm_parameters(  
  scores,  
  angles,  
  prefix = "",  
  suffix = "",  
  e_label = "Elev",  
  x_label = "Xval",  
  y_label = "Yval",  
  a_label = "Ampl",  
  d_label = "Disp",  
  f_label = "Fit"  
)
```

Arguments

scores	Required. A numeric vector (or single row data frame) containing one score for each of a set of circumplex scales.
angles	Required. A numeric vector containing the angular displacement of each circumplex scale included in scores (in degrees).
prefix	Optional. A string to append to the beginning of all of the SSM parameters' variable names (default = "").
suffix	Optional. A string to append to the end of all of the SSM parameters' variable names (default = "").
e_label	Optional. A string representing the variable name of the SSM elevation parameter (default = "Elev").
x_label	Optional. A string representing the variable name of the SSM x-value parameter (default = "Xval").
y_label	Optional. A string representing the variable name of the SSM y-value parameter (default = "Yval").
a_label	Optional. A string representing the variable name of the SSM amplitude parameter (default = "Ampl").
d_label	Optional. A string representing the variable name of the SSM displacement parameter (default = "Disp").
f_label	Optional. A string representing the variable name of the SSM fit or R-squared value (default = "Fit").

Value

A tibble containing the SSM parameters calculated from scores.

See Also

Other ssm functions: [ssm_analyze\(\)](#), [ssm_append\(\)](#), [ssm_plot\(\)](#), [ssm_score\(\)](#), [ssm_table\(\)](#)

Other analysis functions: [ssm_analyze\(\)](#), [ssm_score\(\)](#)

Examples

```
# Manually enter octant scores
scores <- c(0.55, 0.58, 0.62, 0.76, 1.21, 1.21, 1.48, 0.90)
ssm_parameters(scores, angles = octants())

# Customize several of the labels
ssm_parameters(scores, angles = octants(), x_label = "LOV", y_label = "DOM")

# Add a prefix to all labels
ssm_parameters(scores, angles = octants(), prefix = "IIP_")
```

ssm_plot	<i>Create a figure from SSM results</i>
----------	---

Description

Take in the results of an SSM analysis function and create figure from it.

Usage

```
ssm_plot(.ssm_object, fontsize = 12, ...)
```

Arguments

<code>.ssm_object</code>	Required. The results output of ssm_analyze .
<code>fontsize</code>	Optional. A single positive number indicating the font size of text in the figure, in points (default = 12).
<code>...</code>	Additional arguments to pass on to the plotting function.

Value

A `ggplot2` object representing the figure

See Also

[ggsave](#) Function for saving plots to image files.

Other `ssm` functions: [ssm_analyze\(\)](#), [ssm_append\(\)](#), [ssm_parameters\(\)](#), [ssm_score\(\)](#), [ssm_table\(\)](#)

Examples

```
# Load example data
data("jz2017")

# Plot profile results
res <- ssm_analyze(jz2017,
  scales = PA:NO, angles = octants(),
  measures = c(NARPD, ASPD)
)
p <- ssm_plot(res)

# Plot contrast results
res <- ssm_analyze(jz2017,
  scales = PA:NO, angles = octants(),
  measures = c(NARPD, ASPD), contrast = "test"
)
p <- ssm_plot(res)
```

ssm_plot_circle

Create a Circular Plot of SSM Results

Description

Take in the results of a Structural Summary Method analysis and plot the point and interval estimate for each row (e.g., group or measure) in a circular space quantified by displacement and amplitude.

Usage

```
ssm_plot_circle(
  .ssm_object,
  amax = NULL,
  legend_font_size = 12,
  scale_font_size = 12,
  lowfit = TRUE,
  repel = FALSE,
  angle_labels = NULL,
  legend.box.spacing = 0,
  palette = "Set2",
  ...
)
```

Arguments

<code>.ssm_object</code>	The output of <code>ssm_profiles()</code> or <code>ssm_measures()</code> .
<code>amax</code>	A positive real number corresponding to the radius of the circle. It is used to scale the amplitude values and will determine which amplitude labels are drawn.
<code>legend_font_size</code>	A positive real number corresponding to the size (in pt) of the text labels in the legend (default = 12).
<code>scale_font_size</code>	A positive real number corresponding to the size (in pt) of the text labels for the amplitude and displacement scales (default = 12).
<code>lowfit</code>	A logical determining whether profiles with low model fit (<.70) should be plotted, with dashed borders (default = TRUE).
<code>repel</code>	An experimental argument for plotting text labels instead of colors.
<code>angle_labels</code>	A character vector specifying text labels to plot around the circle for each scale. Can also specify NULL to default to numerical angle labels or a vector of empty strings ("") to hide the labels. If not NULL, must have the same length and ordering as the angles argument to <code>ssm_analyze()</code> . (default = NULL)
<code>legend.box.spacing</code>	A double corresponding to the distance (in inches) to add between the data plot and the legend (default = 0).

palette	A string corresponding to the palette to be used from ColorBrewer for the color and fill aesthetics. If set to NULL, all points will appear blue and no legend will be there (useful for showing the coverage of a high number of variables).
...	Currently ignored.

Value

A ggplot variable containing a completed circular plot.

ssm_plot_contrast	<i>Create a Difference Plot of SSM Contrast Results</i>
-------------------	---

Description

Take in the results of a Structural Summary Method analysis with pairwise contrasts and plot the point and interval estimates for each parameter's contrast (e.g., between groups or measures).

Usage

```
ssm_plot_contrast(
  .ssm_object,
  axislabel = "Difference",
  xy = TRUE,
  color = "red",
  linesize = 1.25,
  fontsize = 12
)
```

Arguments

.ssm_object	Required. The results output of ssm_analyze.
axislabel	Optional. A string to label the y-axis (default = "Difference").
xy	A logical determining whether the X-Value and Y-Value parameters should be included in the plot (default = TRUE).
color	Optional. A string corresponding to the color of the point range (default = "red").
linesize	Optional. A positive number corresponding to the size of the point range elements in mm (default = 1.5).
fontsize	Optional. A positive number corresponding to the size of the axis labels, numbers, and facet headings in pt (default = 12).

Value

A ggplot variable containing difference point-ranges faceted by SSM parameter. An interval that does not contain the value of zero has $p < .05$.

ssm_plot_scores	<i>Create a spider/radar plot of circumplex scores</i>
-----------------	--

Description

Create a spider/radar plot of circumplex scores, either from a data frame containing scale scores or the result of `ssm_analyze()`.

Usage

```
ssm_plot_scores(  
  x,  
  amin = NULL,  
  amax = NULL,  
  angle_labels = NULL,  
  linewidth = 1,  
  pointsize = 3,  
  ...  
)
```

Arguments

<code>x</code>	A dataframe or ssm result object.
<code>amin</code>	An optional number to set as the minimum amplitude (center of circle). If set to NULL, will try to detect a reasonable value.
<code>amax</code>	An optional number to set as the maximum amplitude (outer ring of circle). If set to NULL, will try to detect a reasonable value.
<code>angle_labels</code>	An optional character vector to display outside the circle at each angle. Must be the same length as the number of angles.
<code>linewidth</code>	An optional width for the lines of the profile polygons.
<code>pointsize</code>	An optional size for the points at the scale scores.
<code>...</code>	Additional arguments for the S3 methods

Value

A spider/radar plot object

ssm_score	<i>Calculate SSM parameters by row and add results as new columns</i>
-----------	---

Description

Calculate the SSM parameters for each row of a data frame and add the results as additional columns. This can be useful when the SSM is being used for the description or visualization of individual data points rather than for statistical inference on groups of data points.

Usage

```
ssm_score(.data, scales, angles, ...)
```

Arguments

<code>.data</code>	Required. A data frame containing at least circumplex scales.
<code>scales</code>	Required. The variable names or column numbers for the variables in <code>.data</code> that contain circumplex scales to be analyzed.
<code>angles</code>	Required. A numeric vector containing the angular displacement of each circumplex scale included in <code>scales</code> (in degrees).
<code>...</code>	Optional. Additional parameters to pass to <code>ssm_parameters()</code> , such as <code>prefix</code> and <code>suffix</code> .

Value

A data frame containing `.data` plus six additional columns containing the SSM parameters (calculated rowwise).

See Also

Other ssm functions: `ssm_analyze()`, `ssm_append()`, `ssm_parameters()`, `ssm_plot()`, `ssm_table()`

Other analysis functions: `ssm_analyze()`, `ssm_parameters()`

Examples

```
data("aw2009")
ssm_score(aw2009, scales = PA:NO, angles = octants())
```

ssm_table	<i>Create HTML table from SSM results or contrasts</i>
-----------	--

Description

Take in the results of an SSM analysis and return an HTML table with the desired formatting.

Usage

```
ssm_table(.ssm_object, caption = NULL, xy = TRUE, render = TRUE)
```

Arguments

<code>.ssm_object</code>	The output of <code>ssm_profiles()</code> or <code>ssm_measures()</code>
<code>caption</code>	A string to be displayed above the table (default = <code>NULL</code>).
<code>xy</code>	A logical indicating whether the x-value and y-value parameters should be included in the table as columns (default = <code>TRUE</code>).
<code>render</code>	A logical indicating whether the table should be displayed in the RStudio viewer or web browser (default = <code>TRUE</code>).

Value

A tibble containing the information for the HTML table. As a side-effect, may also output the HTML table to the web viewer.

See Also

Other ssm functions: [ssm_analyze\(\)](#), [ssm_append\(\)](#), [ssm_parameters\(\)](#), [ssm_plot\(\)](#), [ssm_score\(\)](#)

Other table functions: [html_render\(\)](#), [ssm_append\(\)](#)

Examples

```
# Load example data
data("jz2017")

# Create table of profile results
res <- ssm_analyze(jz2017,
  scales = PA:NO, angles = octants(),
  measures = c(NARPD, ASPD)
)
ssm_table(res)

# Create table of contrast results
res <- ssm_analyze(jz2017,
  scales = PA:NO, angles = octants(),
  measures = c(NARPD, ASPD), contrast = "test"
)
```

```
ssm_table(res)
```

```
standardize
```

```
Standardize circumplex scales using normative data
```

Description

Take in a data frame containing circumplex scales, angle definitions for each scale, and normative data (from the package or custom) and return that same data frame with each specified circumplex scale transformed into standard scores (i.e., z-scores) based on comparison to the normative data.

Usage

```
standardize(
  .data,
  scales,
  angles,
  instrument,
  sample = 1,
  prefix = "",
  suffix = "_z"
)
```

Arguments

<code>.data</code>	Required. A data frame containing at least circumplex scales.
<code>scales</code>	Required. The variable names or column numbers for the variables in <code>.data</code> that contain circumplex scales to be standardized.
<code>angles</code>	Required. A numeric vector containing the angular displacement of each circumplex scale included in <code>scales</code> (in degrees).
<code>instrument</code>	Required. An instrument object from the package. To see the available circumplex instruments, see <code>instruments()</code> .
<code>sample</code>	Required. An integer corresponding to the normative sample to use in standardizing the scale scores (default = 1). See <code>?norms</code> to see the normative samples available for an instrument.
<code>prefix</code>	Optional. A string to include at the beginning of the newly calculated scale variables' names, before the scale name and <code>suffix</code> (default = "").
<code>suffix</code>	Optional. A string to include at the end of the newly calculated scale variables' names, after the scale name and <code>prefix</code> (default = "_z").

Value

A data frame that matches `.data` except that new variables are appended that contain standardized versions of scales. These new variables will have the same name as `scales` but with a "_z" suffix.

See Also

Other tidying functions: [ipsatize\(\)](#), [score\(\)](#)

Examples

```
data("jz2017")
instrument("iipsc")
standardize(jz2017, PA:NO, octants(), instrument = iipsc, sample = 1)
```

Index

- * **analysis functions**
 - ssm_analyze, 13
 - ssm_parameters, 16
 - ssm_score, 22
 - * **datasets**
 - aw2009, 3
 - jz2017, 7
 - raw_iipsc, 11
 - * **instrument functions**
 - anchors, 2
 - instrument, 4
 - instruments, 5
 - items, 7
 - norms, 8
 - scales, 11
 - * **ssm functions**
 - ssm_analyze, 13
 - ssm_append, 15
 - ssm_parameters, 16
 - ssm_plot, 18
 - ssm_score, 22
 - ssm_table, 23
 - * **table functions**
 - html_render, 4
 - ssm_append, 15
 - ssm_table, 23
 - * **tidying functions**
 - ipsatize, 6
 - score, 12
 - standardize, 24
 - * **visualization functions**
 - ssm_plot, 18
- anchors, 2, 5, 7, 9, 11
- aw2009, 3
- html_render, 4, 16, 23
- instrument, 3, 4, 5, 7, 9, 11
- instruments, 3, 5, 5, 7, 9, 11
- ipsatize, 6, 12, 25
- items, 3, 5, 7, 9, 11
- jz2017, 7
- norms, 3, 5, 7, 8, 11
- octants, 9
- poles, 10
- quadrants, 10
- raw_iipsc, 11
- scales, 3, 5, 7, 9, 11
- score, 6, 12, 25
- ssm_analyze, 13, 16–18, 22, 23
- ssm_append, 4, 14, 15, 17, 18, 22, 23
- ssm_parameters, 14, 16, 16, 18, 22, 23
- ssm_plot, 14, 16, 17, 18, 22, 23
- ssm_plot_circle, 19
- ssm_plot_contrast, 20
- ssm_plot_scores, 21
- ssm_score, 14, 16–18, 22, 23
- ssm_table, 4, 14, 16–18, 22, 23
- standardize, 6, 12, 24