Package 'circumplex'

October 28, 2024

Type Package

Title Analysis and Visualization of Circular Data

Version 1.0.0

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> 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 boot (>= 1.3-18), ggforce (>= 0.3.0), ggplot2 (>= 3.3.0),
htmlTable (>= 1.13.3), Rcpp, rlang, stats

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), vdiffr

LinkingTo Rcpp, RcppArmadillo (>= 0.11)

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

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 2024-10-28 04:30:02 UTC

Contents

---

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 respondants 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: instrument(), instruments(), 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

---

html_render   *Format and render data frame as HTML table*

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 htmlTable.

Value

HTML syntax for the df table.

See Also

Other table functions: [ssm_table](#)()

---

instrument   *Load a specific instrument object*

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., iip32). The code of the instrument assigned by this package and displayed in parentheses by instruments().

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")
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, prefix = "", suffix = "_i", append = TRUE)
```

Arguments

data	Required. A data frame or matrix containing at least circumplex scales.
items	Required. A character vector containing the column names, or a numeric vector containing column indexes, of item variables in data to be ipsatized.
na.rm	Optional. A logical that determines whether missing values should be ignored during the calculation of the mean during ipsatization (default = TRUE).
prefix	Optional. A string that will be added to the start of each items name in the output (default = "").
suffix	Optional. A string that will be added to the end of each items name in the output (default = "_i").
append	Optional. A logical that determines whether to append the ipsatized scores to data in the output or just return the ipsatized scores alone (default = TRUE).

Value

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

See Also

Other tidying functions: [norm_standardize](), [score]()

Examples

```
data("raw_iipsc")
ipsatize(raw_iipsc, items = 1:32)
ipsatize(raw_iipsc, items = sprintf("IIP%02d", 1:32))
```

---

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](), [instrument](), [instruments](), [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

---

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(), instrument(), instruments(), items(), scales()

Examples

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

---

norm_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

```
norm_standardize(
  data,
  scales,
  angles = octants(),
  instrument,
  sample = 1,
  prefix = "",
  suffix = "_z",
  append = TRUE
)
```

Arguments

data	Required. A data frame or matrix containing at least circumplex scales.
scales	Required. A character vector containing the column names, or a numeric vector containing the column indexes, for the variables (scale scores) to be standardized.
angles	Required. A numeric vector containing the angular displacement of each circumplex scale included in scales (in degrees). Can use the octants(), poles(), or quadrants() convenience functions.

instrument	Required. An instrument object from the package. To see the available circumplex instruments, see instruments().
sample	Required. An integer corresponding to the normative sample to use in standardizing the scale scores (default = 1). See ?norms to see the normative samples available for an instrument.
prefix	Optional. A string to include at the beginning of the newly calculated scale variables' names, before the scale name and suffix (default = "").
suffix	Optional. A string to include at the end of the newly calculated scale variables' names, after the scale name and prefix (default = "_z").
append	Optional. A logical that determines whether the calculated standardized scores should be added as columns to data in the output or the standardized scores alone should be output (default = TRUE).

Value

A data frame that contains the norm-standardized versions of scales.

See Also

Other tidying functions: ipsatize(), score()

Examples

```
data("jz2017")
instrument("iipsc")
norm_standardize(jz2017, scales = 2:9, instrument = iipsc, sample = 1)
```

---

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()
```

PANO *Two-letter abbreviations for octant circumplex scales*

Description

Return a vector of abbreviations for octant circumplex scales, from PA to NO.

Usage

```
PANO()
```

Value

A character vector with eight elements, each corresponding to the abbreviation of an octant sub-scale: PA, BC, DE, FG, HI, JK, LM, NO.

Examples

```
PANO()
```

---

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   *Raw item responses on real circumplex scales*

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   *Display the scales of a circumplex instrument*

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()](), [instrument()](), [instruments()](), [items()](), [norms()]()

Examples

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

---

score   *Score circumplex scales from item responses*

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 = "",
  append = TRUE
)
```

Arguments

data	Required. A data frame containing at least circumplex scales.
items	Required. The variable names or column numbers for the variables in .data that contain all the circumplex items from a single circumplex measure, in ascending order from item 1 to item N.
instrument	Required. An instrument object from the package. To see the available circumplex instruments, use instruments().
na.rm	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.
prefix	Optional. A string to include at the beginning of the newly calculated scale variables' names, before Abbrev from key and suffix (default = "").
suffix	Optional. A string to include at the end of the newly calculated scale variables' names, after Abbrev from key and prefix (default = "").
append	Optional. A logical that determines whether the calculated score variables will be appended to data or returned on their own (default = TRUE).

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](), [norm_standardize]()

Examples

```
data("raw_iipsc")
instrument("iipsc")
score(raw_iipsc, items = 1:32, instrument = iipsc, prefix = "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 = FALSE,
  boots = 2000,
  interval = 0.95,
  listwise = TRUE,
  measures_labels = NULL
)
```

Arguments

data	Required. A data frame containing at least circumplex scales.
scales	Required. A character vector of column names, or a numeric vector of column indexes, from data that contains the circumplex scale scores to be analyzed.
angles	Optional. A numeric vector containing the angular displacement of each circumplex scale included in scales (in degrees). (default = octants()).
measures	Optional. Either NULL or a character vector of column names from data that contains one or more variables to be correlated with the circumplex scales and analyzed using correlation-based SSM analyses.
grouping	Optional. Either NULL or a string that contains the column name from data of the variable that indicates the group membership of each observation.
contrast	Optional. A logical indicating whether to output the difference between two measures' or two groups' SSM parameters. Can only be set to TRUE when there are exactly two measures and one group, one measure and two groups, or no measures and two groups (default = FALSE).
boots	Optional. A single positive whole number 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 list-
                  wise 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. Either NULL or 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.

Value

A list containing the results and description of the analysis.

results   A data frame 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 data frame containing the mean scale scores

type   A string indicating what type of SSM analysis was done

See Also

Other ssm functions: ssm_parameters(), 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 = c("PA", "BC", "DE", "FG", "HI", "JK", "LM", "NO")
)

# Single-group correlation-based SSM
ssm_analyze(
  jz2017,
  scales = c("PA", "BC", "DE", "FG", "HI", "JK", "LM", "NO"),
  measures = c("NARPD", "ASPD")
)

# Multiple-group mean-based SSM
ssm_analyze(
  jz2017,
  scales = c("PA", "BC", "DE", "FG", "HI", "JK", "LM", "NO"),
  grouping = "Gender"
)
```

```
# Multiple-group mean-based SSM with contrast
ssm_analyze(
  jz2017,
  scales = c("PA", "BC", "DE", "FG", "HI", "JK", "LM", "NO"),
  grouping = "Gender",
  contrast = TRUE
)

# Single-group correlation-based SSM with contrast
ssm_analyze(
  jz2017,
  scales = c("PA", "BC", "DE", "FG", "HI", "JK", "LM", "NO"),
  measures = c("NARPD", "ASPD"),
  contrast = TRUE
)

# Multiple-group correlation-based SSM
ssm_analyze(
  jz2017,
  scales = c("PA", "BC", "DE", "FG", "HI", "JK", "LM", "NO"),
  measures = "NARPD",
  grouping = "Gender"
)

# Multiple-group correlation-based SSM with contrast
ssm_analyze(
  jz2017,
  scales = c("PA", "BC", "DE", "FG", "HI", "JK", "LM", "NO"),
  measures = "NARPD",
  grouping = "Gender",
  contrast = TRUE
)
```

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 data frame 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 = octants(),
```

```
    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 data frame containing the SSM parameters calculated from scores.

See Also

Other ssm functions: ssm_analyze(), 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)

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

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

---

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,
  drop_lowfit = FALSE,
  repel = FALSE,
  angle_labels = NULL,
  legend.box.spacing = 0,
  palette = "Set2",
  ...
)
```

Arguments

ssm_object	Required. The output of ssm_analyze().
amax	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.
legend_font_size	
	A positive real number corresponding to the size (in pt) of the text labels in the legend (default = 12).
scale_font_size	
	A positive real number corresponding to the size (in pt) of the text labels for the amplitude and displacement scales (default = 12).
drop_lowfit	A logical determining whether profiles with low model fit (<.70) should be omitted or plotted with dashed borders (default = FALSE).
repel	An experimental argument for plotting text labels instead of colors.

angle_labels   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 ssm_analyze(). (default = NULL)

legend.box.spacing
                  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.

Examples

```
data("jz2017")
res <- ssm_analyze(
  jz2017,
  scales = 2:9,
  measures = c("NARPD", "ASPD")
)
ssm_plot_circle(res)
```

---

ssm_plot_contrast   *Create a Difference Plot of SSM Contrast Results*

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,
  drop_xy = FALSE,
  sig_color = "#fc8d62",
  ns_color = "white",
  linesize = 1.25,
  fontsize = 12,
  ...
)
```

Arguments

`ssm_object`	Required. The results output of `ssm_analyze()`.
`drop_xy`	A logical determining whether the X-Value and Y-Value parameters should be removed from the plot (default = FALSE).
`sig_color`	Optional. A string corresponding to the color to use to denote significant contrasts (default = "#fc8d62").
`ns_color`	Optional. A string corresponding to the color to use to denote non-significant contrasts (default = "white").
`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).
`...`	Additional arguments will be ignored.

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.

Examples

```
data("jz2017")
res <- ssm_analyze(
  jz2017,
  scales = 2:9,
  measures = c("NARPD", "ASPD"),
  contrast = TRUE
)
ssm_plot_contrast(res)
```

---

ssm_plot_curve	*Create a Curve Plot of SSM Results*

Description

Take in the results of a Structural Summary Method analysis and plot the scores by angle and the estimated SSM curve.

Usage

```
ssm_plot_curve(
  ssm_object,
  angle_labels = NULL,
  base_size = 11,
  drop_lowfit = FALSE,
  ...
)
```

Arguments

`ssm_object`	Required. The results output of ssm_analyze().
`angle_labels`	Optional. Either NULL or a character vector that determines the x-axis labels. If NULL, the labels will be the angle numbers. If a character vector, must be the same length and in the same order as the angles argument to ssm_analyze() (default = NULL).
`base_size`	Optional. A positive number corresponding to the base font size in pts (default = 11).
`drop_lowfit`	Optional. A logical indicating whether to omit profiles with low fit (<.70) or include them with dashed lines (default = FALSE).
`...`	Additional arguments will be ignored.

Value

A ggplot object depicting the SSM curve(s) of each profile.

Examples

```
data("jz2017")
res <- ssm_analyze(
  jz2017,
  scales = 2:9,
  measures = 10:13
)
ssm_plot_curve(res)
ssm_plot_curve(res, angle_lables = PANO())
```

---

ssm_score   *Calculate SSM parameters by row and add results as new columns*

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 = octants(), append = TRUE, ...)
```

Arguments

data	Required. A data frame containing at least circumplex scales.
scales	Required. The variable names or column numbers for the variables in `.data` that contain circumplex scales to be analyzed.
angles	Required. A numeric vector containing the angular displacement of each circumplex scale included in `scales` (in degrees).
append	Optional. A logical indicating whether to append the output to `data` or simply return the output (default = "TRUE").
...	Optional. Additional parameters to pass to `ssm_parameters()`, such as `prefix` and `suffix`.

Value

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

See Also

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

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

Examples

```
data("aw2009")
ssm_score(
  aw2009,
  scales = c("PA", "BC", "DE", "FG", "HI", "JK", "LM", "NO")
)
```

---

ssm_table   *Create HTML table from SSM results or contrasts*

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, drop_xy = FALSE, render = TRUE)
```

Arguments

`ssm_object`	Required. The results output of `ssm_analyze()`.
`caption`	A string to be displayed above the table (default = NULL).
`drop_xy`	A logical indicating whether the x-value and y-value parameters should be omitted from the output (default = FALSE).
`render`	A logical indicating whether the table should be displayed in the RStudio viewer or web browser (default = TRUE).

Value

A data frame 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_score()`

Other table functions: `html_render()`

Examples

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

# Create table of profile results
res <- ssm_analyze(
  jz2017,
  scales = 2:9,
  measures = c("NARPD", "ASPD")
)
ssm_table(res)

# Create table of contrast results
res <- ssm_analyze(
  jz2017,
  scales = 2:9,
  measures = c("NARPD", "ASPD"),
  contrast = TRUE
)
ssm_table(res)
```

Index