Package: WJSmisc (via r-universe)

November 8, 2024

Title Miscellaneous functions from W. Joel Schneider

Date 2021-03-10

Version 0.3

Description Several functions I find useful.

License CC0

URL https://github.com/wjschne/WJSmisc

BugReports https://github.com/wjschne/WJSmisc/issues

Imports corrr, dplyr, forcats, ggplot2, magrittr, patchwork, psych, rlang (>= 0.1.2), stats, grDevices, stringr, superheat, rstudioapi, shiny, tibble, tidyr, GPArotation, utils, purrr, rlist, scales, readr, tidyselect, ggtext, signs

Encoding UTF-8

RoxygenNote 7.2.3

Suggests roxygen2, testthat (>= 2.1.0)

Config/pak/sysreqs make libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

Repository https://wjschne.r-universe.dev

RemoteUrl https://github.com/wjschne/WJSmisc

RemoteRef HEAD

RemoteSha 85d58415775ff5f004d101ad2478e4c2b02fe8ba

Contents

`angle2hjust` *Convert angles to ggplot2 hjust*

Description

Convert angles to ggplot2 hjust

Usage

```
angle2hjust(theta, multiplier = 1.5, as_degrees = FALSE)
```

Arguments

`theta`	angle in radians
`multiplier`	distance from point
`as_degrees`	use degrees instead of radians

Value

numeric

`angle2vjust` *Convert angles to ggplot2 vjust*

Description

Convert angles to ggplot2 vjust

Usage

```
angle2vjust(theta, multiplier = 1.5, as_degrees = FALSE)
```

Arguments

`theta`	angle in radians
`multiplier`	distance from point
`as_degrees`	use degrees instead of radians

Value

numeric

Examples

```
library(tibble)
library(ggplot2)
library(dplyr)
xy_ratio <- pi
tibble(theta = seq(0,2*pi, length.out = 9),
       y =  sin(theta),
       slope = cos(theta) * xy_ratio,
       text_angle = atan(slope) + pi / 2) %>%
   ggplot(aes(theta,y)) +
     stat_function(fun = sin) +
     geom_point() +
     geom_label(aes(label = LETTERS[1:9],
                    vjust = angle2vjust(text_angle, multiplier = 1.5),
                    hjust = angle2hjust(text_angle, multiplier = 1.5)),
                label.size = NA,
                label.padding = unit(1, "mm")) +
  scale_x_continuous(expression(theta),
                     breaks = seq(0,2*pi, length.out = 9),
                    labels = label_parsed(c("0", "frac(pi,4)", "frac(pi,2)",
                                            "frac(3 * pi,4)", "pi", "frac(5*pi,4)",
                                            "frac(3 * pi,2)", "frac(7*pi,4)", "2*pi"))) +
  scale_y_continuous(expression(sin(theta))) +
  coord_fixed(ratio = xy_ratio, clip = "off") +
  theme_minimal()
```

`attach_function` *Converts the default values of a function's arguments to variables and attaches them to the global environment*

Description

Converts the default values of a function's arguments to variables and attaches them to the global environment

Usage

```
attach_function(f)
```

Arguments

`f`	Function

Value

Attaches function arguments to global environment

Examples

```
my_function <- function(x, y = 2) x + y

# Sets y to 2
attach_function(my_function)
```

`cor2pcor` *Converts a correlation matrix to a partial correlation matrix*

Description

Converts a correlation matrix to a partial correlation matrix

Usage

```
cor2pcor(R)
```

Arguments

`R`	correlation matrix

Value

Partial correlation matrix

Examples

```
R <- matrix(.6, nrow = 3, ncol = 3)
diag(R) <- 1
cor2pcor(R)
```

`cornames<-` *Name for square arrays like correlation matrices*

Description

Name for square arrays like correlation matrices

Usage

```
cornames(x) <- value
```

Arguments

`x`	a square array, matrix, or data.frame
`value`	a vector of names

Value

a named square array, matrix, or data.frame

Examples

```
R <- tri2cor(.5)
cornames(R) <- c("A", "B")
```

`cor_ellipse` *Generate correlation ellipse data*

Description

Generate correlation ellipse data

Usage

```
cor_ellipse(
  r = 0,
  mean = c(0, 0),
  sd = c(1, 1),
  p = 0.95,
  split_x = NULL,
  split_y = NULL,
  n_points = 10000
)
```

Arguments

`r`	Correlation coefficient
`mean`	Vector of 2 means
`sd`	Vector of 2 standard deviations
`p`	Proportion of data ellipse covers
`split_x`	Split ellipse at x value
`split_y`	Split ellipse at y value
`n_points`	Number of points

Value

data.frame

Examples

```
cor_ellipse(r = .75)
```

`cor_heat` *Correlation plot*

Description

Correlation plot

Usage

```
cor_heat(
  d,
  test_names = colnames(d),
  margins = 0.55,
  text_size = 4,
  dendrograms = TRUE,
  palette_col = c("firebrick", "white", "royalblue"),
  x_label_angle = 90,
  reorder_vars = TRUE,
  heat.lim = c(-1, 1),
  heat.pal.values = seq(0, 1, 1/(length(palette_col) - 1)),
  ...
)
```

Arguments

`d`	Data or correlation matrix
`test_names`	Vector of names of variables
`margins`	Width of margins for labels
`text_size`	Size of text
`dendrograms`	If TRUE, add dendrograms
`palette_col`	A vector of three colors for heatmap
`x_label_angle`	Angle of x-axis label
`reorder_vars`	If TRUE, reorder variables based on cluster analysis
`heat.lim`	Vector of the lower and upper bounds on heat map
`heat.pal.values`	Vector of values of where on the scale each color in palette_col falls. Defaults to even spacing.
`...`	Additional parameters passed to superheat::superheat

`cor_text` *Format text of a correlation matrix*

Description

Format text of a correlation matrix

Usage

```
cor_text(r, digits = 2, remove_leading_zero = TRUE, remove_diagonal = FALSE)
```

Arguments

`r`	a matrix of correlations
`digits`	Number of digits to round
`remove_leading_zero`	If TRUE, remove leading zero from all correlations
`remove_diagonal`	If TRUE, remove diagonal of ones

Value

a matrix of correlations as text

Examples

```
cor_text(matrix(.5,2,2))
```

`cross_vectors` *Create unique combinations of vectors*

Description

Create unique combinations of vectors

Usage

```
cross_vectors(..., sep = "_")
```

Arguments

`...`	vectors
`sep`	Separate character

Value

A character vector

Examples

```
cross_vectors(c("a", "b"), c(1,2))
```

df2matrix *Convert data.frame and tibbles to matrices with named rows and columns*

Description

Convert data.frame and tibbles to matrices with named rows and columns

Usage

```
df2matrix(d, first_col_is_row_names = TRUE)
```

Arguments

`d`	data.frame or tibble
`first_col_is_row_names`	TRUE if first column has row names

Value

matrix

Examples

```
d <- data.frame(rname = c("x1", "x2"), x1 = c(1,2), x2 = c(3,4))
df2matrix(d)
```

geom_richlabel *A wrapper for ggtext::geom_richtext*

Description

A wrapper for ggtext::geom_richtext

Usage

```
geom_richlabel(
  mapping,
  label.margin = unit(2, "mm"),
  label.padding = unit(0.5, "mm"),
  label.color = NA,
  fill = "white",
  text_size = 12,
  ...
)
```

Arguments

`mapping`	aes()
`label.margin`	grid unit margin
`label.padding`	grid unit margin
`label.color`	color
`fill`	color
`text_size`	text size in point units
`...`	additional parameters passed to ggtext::geom_richtext

Value

ggtext::geom_richtext

Examples

```
library(tibble)
library(ggplot2)
library(dplyr)
library(WJSmisc)
tibble(x = 0, y = 0, l = "A") %>%
  ggplot() +
  geom_richlabel(aes(x,y,label = l))
```

`get_quote` *Retrieve text within a span or div with a named id*

Description

Retrieve text within a span or div with a named id

Usage

```
get_quote(id, file, blockquote = TRUE)
```

Arguments

`id`	the id string (without #).
`file`	file name
`blockquote`	return as block quote (defaults to TRUE)

Value

a character vector of length 1

ggsvg *Save ggplot as .pdf, then convert to .svg via pdf2svg*

Description

Save ggplot as .pdf, then convert to .svg via pdf2svg

Usage

```
ggsvg(f, width = 6.5, height = 6.5, ...)
```

Arguments

`f`	Filename of the svg file. Omit the ".svg" at the end.
`width`	width passed to ggplot2::ggsave
`height`	width passed to ggplot2::ggsave
`...`	Additional parameters passed to ggplot2::ggsave

ggtext_size *Convert ggplot theme font size to geom_text size*

Description

Convert ggplot theme font size to geom_text size

Usage

```
ggtext_size(base_size, ratio = 0.8)
```

Arguments

`base_size`	theme font size
`ratio`	ratio of text size to theme font size. Defaults to .8 so that geom text will be the same size as default sized axis labels.

Value

numeric vector

Examples

```
ggtext_size(16)
```

insert_latex_color *Insert latex colors into a matrix*

Description

Insert latex colors into a matrix

Usage

```
insert_latex_color(
  m,
  color_cells,
  color_rownames = NULL,
  color_colnames = NULL
)
```

Arguments

`m`	matrix of values
`color_cells`	matrix of latex colors for matrix cells
`color_rownames`	vector of latex colors for row names
`color_colnames`	vector of latex colors for column names

Value

matrix

Examples

```
# A matrix of zeros
m <- matrix(0, nrow = 2, ncol = 2)

# A matrix of NA values the same size as m
latex_colors <- m * NA

# Make the top row red
latex_colors[1,] <- "red"

# Insert colors into m
insert_latex_color(m, latex_colors)
```

`irt_plot_app` *IRT Plot Shiny App*

Description

IRT Plot Shiny App

Usage

```
irt_plot_app()
```

`latexarray` *Make latex array*

Description

Make latex array

Usage

```
latexarray(
  M,
  left = "",
  right = "",
  env = "array",
  includenames = TRUE,
  align = NULL,
  lines = TRUE
)
```

Arguments

`M`	A matrix
`left`	left delimiter options: [(|
`right`	right delimiter options:])|
`env`	Array environment
`includenames`	Include column and row names
`align`	Column alignment. For a three column matrix, alignment defaults to centering columns("ccc"). If there are row labels, the default would be "r|ccc" to right-align the row labels and separate them with vertical line.
`lines`	Include lines separating column and row names

Value

character string

Examples

```
M <- diag(3)
colnames(M) <- LETTERS[1:3]
rownames(M) <- LETTERS[1:3]
latexarray(M)
```

lm_matrix *Regression from correlation matrix*

Description

Regression from correlation matrix

Usage

```
lm_matrix(R, ind, dep)
```

Arguments

R	correlation matrix
ind	independent variables
dep	dependent variable

Value

list of coefficients and R2

make_indicators *Make latent variable indicators with rbeta_ms function*

Description

Make latent variable indicators with rbeta_ms function

Usage

```
make_indicators(
  latent,
  indicators = NULL,
  mu = 0.8,
  sigma = 0.05,
  k = 3,
  digits = 3
)
```

Arguments

`latent`	name of latent variable
`indicators`	vector of indicator names (assigned automatically if left NULL)
`mu`	mean of standardized coefficients
`sigma`	sd of standardized coeficients
`k`	number of indicator variables
`digits`	number of digits to round coefficients

Value

lavaan code for latent variable assignment

Examples

```
make_indicators("depression", mu = 0.8, sigma = 0.05, k = 4)
```

`modregplot` *Simple slopes plot*

Description

generates simple slopes plot from moderated regression equation

Usage

```
modregplot(
  predictor_range = c(-4, 4),
  moderator_values = c(-1, 0, 1),
  intercept = 0,
  predictor_coef = 0,
  moderator_coef = 0,
  interaction_coef = 0,
  predictor_label = "X",
  criterion_label = "Y",
  moderator_label = "Moderator"
)
```

Arguments

`predictor_range`	a length 2 vector of the range of values to be plotted on the predictor variable, Default: c(-4, 4)
`moderator_values`	a vector of moderator values to be plotted, Default: c(-1, 0, 1)
`intercept`	the intercept of the regression equation, Default: 0

`predictor_coef`
: the regression coefficient for the predictor variable, Default: 0

`moderator_coef`
: the regression coefficient for the moderator variable, Default: 0

`interaction_coef`
: the interaction term coefficent, Default: 0

`predictor_label`
: the label for the predictor variable, Default: 'X'

`criterion_label`
: the label for the moderator variable, Default: 'Y'

`moderator_label`
: PARAM_DESCRIPTION, Default: 'Moderator'

Value

a ggplot of the simple slopes

Examples

```
modregplot(
  predictor_range = c(-2, 2),
  moderator_values = c(Low = -1, High = 1),
  intercept = 6,
  predictor_coef = 2,
  moderator_coef = 0,
  interaction_coef = 1,
  predictor_label = "Psychopathy",
  criterion_label = "Aggression",
  moderator_label = "Impulsivity"
)
```

`multivariate_ci` *General a multivariate confidence interval for a set of scores*

Description

General a multivariate confidence interval for a set of scores

Usage

```
multivariate_ci(x, rxx, mu, sigma, ci = 0.95, v_names = names(x))
```

Arguments

`x`	a vector of scores
`rxx`	a vector reliability coefficients
`mu`	a vector means
`sigma`	a covariance matrix
`ci`	confidence level
`v_names`	a vector of names

Value

data.frame

Examples

```
x_wisc <- c(
  vci = 130,
  vsi = 130,
  fri = 70,
  wmi = 130,
  psi = 130
)
rxx_wisc <- c(
  vci = .92,
  vsi = .92,
  fri = .93,
  wmi = .92,
  psi = .88
  )
R_wisc <- ("
  index vci  vsi  fri  wmi  psi
  vci   1.00 0.59 0.59 0.53 0.30
  vsi   0.59 1.00 0.62 0.50 0.36
  fri   0.59 0.62 1.00 0.53 0.31
  wmi   0.53 0.50 0.53 1.00 0.36
  psi   0.30 0.36 0.31 0.36 1.00") |>
    readr::read_tsv() |>
    tibble::column_to_rownames("index") |>
    as.matrix()
 multivariate_ci(
   x = x_wisc,
   rxx = rxx_wisc,
   mu = rep(100, 5),
   sigma = R_wisc * 225
 )
```

`parallel_analysis` *ggplot of parallel analysis from the psych package*

Description

ggplot of parallel analysis from the psych package

Usage

```
parallel_analysis(
  d,
  fm = "pa",
  factor_based = TRUE,
  vcolors = c("firebrick", "royalblue"),
  font_family = "sans",
  ...
)
```

Arguments

`d`	data to be analyzed
`fm`	factor method passed to psych::fa.parallel
`factor_based`	TRUE is factor-based and FALSE is principal component-based
`vcolors`	vector of 2 colors for lines
`font_family`	Name of font
`...`	parameters passed to psych::fa.parallel

Examples

```
d <- psych::bfi[,1:25]
parallel_analysis(d)
```

paste_matrix_from_clipboard

Paste matrix code from clipboard

Description

Paste matrix code from clipboard

Usage

```
paste_matrix_from_clipboard(digits = 2, as_matrix = TRUE)
```

Arguments

`digits`	Number of digits to round
`as_matrix`	Convert to matrix. Defaults to ‘TRUE‘

`pdf2svg` *Function that converts pdf files to svg. Must have the pdf2svg program installed (https://github.com/dawbarton/pdf2svg)*

Description

Function that converts pdf files to svg. Must have the pdf2svg program installed (https://github.com/dawbarton/pd

Usage

```
pdf2svg(f)
```

Arguments

`f` Filename of the pdf file. Omit the ".pdf" at the end.

`plotnorm` *Plot a normal distribution shading below x*

Description

Plot a normal distribution shading below x

Usage

```
plotnorm(
  x = 0,
  mu = 0,
  sigma = 1,
  below = TRUE,
  show_proportion = TRUE,
  show_x = TRUE,
  show_param = TRUE,
  text_size = 14,
  font_family = "sans",
  shade_fill = "royalblue"
)
```

Arguments

`x` number to divide normal distribution

`mu` mean of normal distribution

`sigma` standard deviation of normal distribution

`below` If TRUE, shade lower portion of normal distribution

`show_proportion`
: If TRUE, display proportions

`show_x` If TRUE, display x value

`show_param` If TRUE, display mean and standard deviation

`text_size` Base text size

`font_family` Name of font

`shade_fill` Color of shaded region

Examples

```
plotnorm(90, 100, 15)
```

`plot_loading` *Creats a plot of factor loadings*

Description

Creats a plot of factor loadings

Usage

```
plot_loading(
  f,
  font_family = "sans",
  font_size = 14,
  loading_text_size = font_size * 0.8,
  factor_names = sort(colnames(f$loadings)),
  nudge_loadings = 0.05
)
```

Arguments

`f` output of a factor analysis from the psych::fa function

`font_family` Name of font

`font_size` Size of font

`loading_text_size`
: size of loading font,

`factor_names` names of the factors #'

`nudge_loadings`
: nudge loadings on x dimension

Examples

```
library(GPArotation)
library(psych)
fit <- fa(psych::bfi[,1:25], nfactors = 5)
plot_loading(fit)
```

prob_label *Format numeric probabilities as text labels*

Description

Format numeric probabilities as text labels

Usage

```
prob_label(
  p,
  accuracy = 0.01,
  digits = NULL,
  max_digits = NULL,
  remove_leading_zero = TRUE,
  round_zero_one = TRUE
)
```

Arguments

`p`	numeric vector of probabilities
`accuracy`	accuracy of rounding
`digits`	Optional. Number of digits to round. Overrides accuracy parameter
`max_digits`	Optional. Maximum zeros or nines before rounding to 0 or 1
`remove_leading_zero`	Removes leading zero from probability
`round_zero_one`	Apply rounding to 0 and 1

Value

character vector

Examples

```
prob_label(seq(0,1, 0.1))
```

`proportion2percentile` *Rounds proportions to significant digits both near 0 and 1, then converts to percentiles*

Description

Rounds proportions to significant digits both near 0 and 1, then converts to percentiles

Usage

```
proportion2percentile(
  p,
  digits = 2,
  remove_leading_zero = TRUE,
  add_percent_character = FALSE
)
```

Arguments

`p` probability

`digits` rounding digits

`remove_leading_zero`
If TRUE, remove leading zero

`add_percent_character`
If TRUE, add percent character

Value

character vector

Examples

```
proportion2percentile(0.011111)
```

`proportion_round` *Rounds proportions to significant digits both near 0 and 1*

Description

Rounds proportions to significant digits both near 0 and 1

Usage

```
proportion_round(p, digits = 2)
```

Arguments

`p`	probabiity
`digits`	rounding digits

Value

numeric vector

Examples

```
proportion_round(0.01111)
```

`pthreshold` *Probability a true score will be below a threshold*

Description

Probability a true score will be below a threshold

Usage

```
pthreshold(x, threshold, rxx, mu = 0, sigma = 1)
```

Arguments

`x`	observed score
`threshold`	threshold
`rxx`	reliability coefficient
`mu`	population mean
`sigma`	population standard deviation

Value

probability

Examples

```
pthreshold(x = .5, threshold = 1, rxx = 0.9)
```

pvalueAPA *APA p-value rounding*

Description

APA p-value rounding

Usage

```
pvalueAPA(p, inline = FALSE, mindigits = 2, maxdigits = 3)
```

Arguments

`p`	probabiity
`inline`	to be used in an inline rmarkdown (default is FALSE)
`mindigits`	minimum rounding digits
`maxdigits`	maximum rounding digits

Value

character

Examples

```
pvalueAPA(0.01111)
```

rbeta_ms *Random beta distribution with specified mean and sd*

Description

Random beta distribution with specified mean and sd

Usage

```
rbeta_ms(n = 1, mu = 0.5, sigma = 0.025)
```

Arguments

`n`	Number of data points
`mu`	Mean of random beta distribution
`sigma`	SD of random beta distribution

Value

a vector of numeric values

Examples

```
rbeta_ms(n = 5, mu = 0.8, sigma = 0.1)
```

`remove_leading_zero` *Remove leading zero from numbers*

Description

Remove leading zero from numbers

Usage

```
remove_leading_zero(x, digits = 2, ...)
```

Arguments

`x`	vector of numbers
`digits`	rounding digits
`...`	Arguments passed to formatC

Value

vector of characters

Examples

```
remove_leading_zero(c(0.5,-0.2))
```

`rotate2dmatrix` *Rotate a 2-column matrix*

Description

Rotate a 2-column matrix

Usage

```
rotate2dmatrix(x, theta, degrees = FALSE, origin = c(0, 0))
```

Arguments

`x`	a 2-column matrix
`theta`	angle
`degrees`	if TRUE, theta is in degrees instead of radians
`origin`	point of rotation

Value

a rotated 2-column matrix

Examples

```
x <- matrix(seq(10), ncol = 2)
rotate2dmatrix(x, pi)
```

signs_centered *centered signed numbers*

Description

A wrapper function for the signs::signs function. It adds a figure space to negative numbers so that it appear as if the minus sign does not affect the number's centering.

Usage

```
signs_centered(x, space = " ", ...)
```

Arguments

`x`	a numeric vector
`space`	a character to be added to negative numbers (defaults to a figure space)
`...`	parameters passed to signs:signs

Value

a vector of numbers converted to characters

Examples

```
library(ggplot2)
d <- data.frame(x = -4:0, y = -4:0)
# In these 2 plots, Compare the centering of the negative numbers on the x-axis
ggplot(d, aes(x,y))
ggplot(d, aes(x,y)) +
  scale_x_continuous(labels = signs_centered)
```

skewed_axis *Draw skewed axes in ggplot2*

Description

Draw skewed axes in ggplot2

Usage

```
skewed_axis(
  theta,
  axis_title = "X",
  draw_ticks = TRUE,
  draw_axis_text = TRUE,
  remove_origin = TRUE,
  tick_height = 0.02,
  lwd = 0.5,
  text_size = 12,
  color = NA,
  family = NA,
  mu = 0,
  sigma = 1,
  tick_label_interval = ifelse(sigma%%3 == 0, 3, 2)
)
```

Arguments

`theta`	angle in radians
`axis_title`	character
`draw_ticks`	logical
`draw_axis_text`	logical
`remove_origin`	logical
`tick_height`	height of ticks
`lwd`	line width
`text_size`	size of text
`color`	color of lines
`family`	font family
`mu`	mean of variable
`sigma`	standard deviation of variable
`tick_label_interval`	interval of ticks

Value

plot

Examples

```
library(ggplot2)
ggplot(data.frame(x = c(0,1), y = c(4,4))) +
  skewed_axis(pi / 2, color = "black") +
  skewed_axis(pi / 4, color = "black") +
  skewed_axis(0, color = "black") +
  theme_void() +
  coord_equal()
```

`snake2subscript` *Convert snake case to subscript*

Description

Convert snake case to subscript

Usage

```
snake2subscript(
  x,
  sep = "_",
  prefix = "<sub>",
  suffix = "</sub>",
  collapse = NULL,
  recycle0 = FALSE
)
```

Arguments

`x`	character
`sep`	separator defaults to "_"
`prefix`	character prefix
`suffix`	character suffix
`collapse`	character parameter passed to paste0
`recycle0`	parameter passed to paste0

Value

a character

Examples

```
snake2subscript("x_1")
```

`str_wrap_equal` *Wrap sentence strings into lines of roughly equal width*

Description

Wrap sentence strings into lines of roughly equal width

Usage

```
str_wrap_equal(x, max_width = 30L, sep = "\n")
```

Arguments

`x`	a character vector
`max_width`	the maximum number of characters in a line (unless a word is longer than ‘max_width‘)
`sep`	character string that separates text lines

Value

a character vector

Examples

```
library(ggplot2)
library(stringr)
library(dplyr)
d <- data.frame(
  Item = c(
    "Lorem ipsum dolor sit amet, consectetur adipiscing elit.",
    "Duis pretium arcu quis nibh elementum, sed aliquam enim dignissim.",
    "Nullam et ornare enim, et egestas odio.",
    "Aliquam posuere ante quis magna rutrum, id elementum nulla sodales.",
    "Interdum et malesuada fames ac ante ipsum primis in faucibus.",
    "Aenean rutrum lorem at metus pretium, malesuada porta tellus facilisis.",
    "Vestibulum at convallis enim.",
    "Nam malesuada bibendum rutrum.",
    "Donec risus sapien, pulvinar vitae porttitor non, lobortis ac felis."
  ),
  Proportion = seq(.1,.9,.1)
) |>
  mutate(Item = forcats::fct_inorder(Item))

# Axis labels with stringr::str_wrap
ggplot(d, aes(Proportion, Item)) +
  geom_col() +
  scale_y_discrete(NULL, labels = \(x) str_wrap(x, width = 25L))

# Axis labels with WJSmisc::str_wrap_equal
```

```
ggplot(d, aes(Proportion, Item)) +
  geom_col() +
  scale_y_discrete(NULL, labels = \(x) str_wrap_equal(x, max_width = 25L))
```

`tri2cor` *Create square correlation matrix from lower triangle*

Description

Create square correlation matrix from lower triangle

Usage

```
tri2cor(x, variable_names = NULL)
```

Arguments

`x` vector of correlations

`variable_names` a vector of variable names

Value

square matrix

Examples

```
tri2cor(c(.2,.3,.4))
```

`vector_angle` *Compute the angle of a vector*

Description

Compute the angle of a vector

Usage

```
vector_angle(x, origin = c(0, 0), degrees = FALSE, allow_negative = FALSE)
```

Arguments

`x` A length-2 vector

`origin` A length-2 vector

`degrees` If TRUE, returns angles in degrees instead of radians

`allow_negative` If TRUE, returns angles between -pi and pi (-180 and +180. If FALSE, returns angles between 0 and 2 * pi (0 and 360)

Value

A length-1 vector

Examples

```
vector_angle(c(1,1))
vector_angle(c(1,1), degrees = TRUE)
```

x2z *Convert x to a z-score*

Description

Convert x to a z-score

Usage

```
x2z(x, mu = mean(x, na.rm = T), sigma = stats::sd(x, na.rm = T))
```

Arguments

x	a numeric vector
mu	mean
sigma	standard deviation

Index