Package: psycheval (via r-universe)

November 8, 2024

Type Package

Title A psychological evaluation toolkit

Version 0.1.0

Description Functions useful in psychological evaluations This package accompanies [*Individual Psychometrics*](https://individual-psychometrics.rbind.io/) online textbook.

License CC0

Encoding UTF-8

LazyData true

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

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

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports lifecycle, stats, WJSmisc (>= 0.3)

Suggests readr, tibble, dplyr, tidyr, ggplot2, testthat (>= 3.0.0)

Config/testthat/edition 3

Remotes https://github.com/wjschne/psycheval

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/psycheval

RemoteRef HEAD

RemoteSha ced533e91ca01a7ff066d72644389826cd288ac9

Contents

`composite_covariance` *Computes covariances of composite scores given a covariance matrix and a weight matrix*

Description

Computes covariances of composite scores given a covariance matrix and a weight matrix

Usage

```
composite_covariance(Sigma, w, correlation = FALSE)
```

Arguments

`Sigma`	Covariance matrix
`w`	Weight matrix. Must have the same number of rows as R
`correlation`	If TRUE, return correlations instead of covariances

Examples

```
# Create variable names
v_names <- c(paste0("A_", 1:3),paste0("B_", 1:3))
v_composites <- c("A", "B")

# Create covariance matrix
Sigma <- matrix(0.6, nrow = 6, ncol = 6, dimnames = list(v_names, v_names))
diag(Sigma) <- 1

# Create weight matrix
w <- matrix(0, nrow = 6, ncol = 2, dimnames = list(v_names, v_composites))
w[v_names[1:3],"A"] <- 1
w[v_names[4:6],"B"] <- 1
w
```

```
# covariance matrix of weighted sums
composite_covariance(Sigma, w)
```

composite_score Compute composite score

Description

Compute composite score

Usage

```
composite_score(
  x,
  R,
  mu_x = 100,
  sigma_x = 15,
  mu_composite = 100,
  sigma_composite = 15,
  w = NULL
)
```

Arguments

x	Vector of subtest scores
R	Subtest score correlation matrix
mu_x	Vector of subtest means
sigma_x	Vector of subtest standard deviations
mu_composite	Composite mean
sigma_composite	Composite standard deviation
w	Vector of weights

Value

composite score

Examples

```
# Subtest scores
x <- c(12, 14)
R <- matrix(c(1,.6, .6, 1), nrow = 2)
composite_score(x = x,
              R = R,
              mu_x = 10,
             sigma_x = 3)
```

conditional_covariance

Conditional Covariance

Description

Conditional Covariance

Usage

```
conditional_covariance(x, sigma, mu = 0)
```

Arguments

x	named numeric vector of predictor scores
sigma	named covariance matrix of predictor and outcome variables
mu	a single numeric mean for all variables or a named vector of means of predictor and outcome variables

Value

list of conditional means and a covariance matrix

- `mu_conditional` - The means of the outcome variables conditioned on the values of the predictors in vector x.
- `mu_sigma` - The covariance matrix of the outcome variables conditioned on the values of the predictors in vector x.
- `descriptives_conditional` - A data frame of means and standard deviations of the outcome variables conditioned on the values of the predictors in vector x.
- `x` - The predictor scores from the x parameter
- `sigma` - The unconditional covariance matrix from the sigma parameter
- `mu` - Anamed vector of unconditional means

Examples

```
# Named vector of predictor scores
x <- c(A = 1)

# Named vector of unconditional means
mu <- c(A = 0, B = 0, C = 0)

# Unconditional covariance matrix with row and column names
sigma <- matrix(c(1, .5, .5,
                  .5, 1, .5,
                  .5, .5, 1),
                nrow = 3,
                ncol = 3,
                dimnames = list(names(mu),
                                names(mu)))

# Conditoinal means and covariance matrix
conditional_covariance(x = x, sigma = sigma, mu = mu)
```

`difference_score` *Difference score statistics*

Description

Difference score statistics

Usage

```
difference_score(
  x,
  y,
  r_xx = 0.9,
  r_yy = 0.9,
  r_xy = 0,
  mu = 100,
  sigma = 15,
  ci = 0.95,
  mu_x = mu,
  mu_y = mu,
  sigma_x = sigma,
  sigma_y = sigma,
  tails = 2
)
```

Arguments

`x`	first score
`y`	second score
`r_xx`	reliability of x
`r_yy`	reliability of y
`r_xy`	correlation between x and y
`mu`	population mean of both x and y
`sigma`	population standard deviation of both x and y
`ci`	confidence interval of difference score
`mu_x`	population mean of x (defaults to mu)
`mu_y`	population mean of y (defaults to mu)
`sigma_x`	population standard deviation of x (defaults to sigma)
`sigma_y`	population standard deviation of y (defaults to sigma)
`tails`	for significance and prevalence of difference scores

Value

list

Examples

```
difference_score(
  x = 120,
  y = 110,
  r_xx = .95,
  r_yy = .92,
  r_xy = .65,
  mu = 100,
  sigma = 15)
```

`logit2w` *Convert logits to W scores*

Description

Convert logits to W scores

Usage

```
logit2w(logit, refw = 500)
```

Arguments

`logit`	numeric vector of logits
`refw`	numeric vector of reference W scores

Value

numeric vector of W scores

Examples

```
logit2w(2)
```

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

Arguments

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

Value

A data frame with the following columns:

- `variable` - Variable names
- `x` - Variable scores
- `r_xx` - Reliability coefficients
- `mu_univariate` - Expected true score estimated from the corresponding observed score
- `see_univariate` - Standard error of the estimate computed from the corresponding reliability coefficient
- `mu_multivariate` - Expected true score estimated from all observed scores
- `see_multivariate` - Standard error of the estimate computed from the corresponding reliability coefficient
- `upper_univariate` - upper bound of univariate confidence interval
- `lower_univariate` - lower bound of univariate confidence interval
- `upper_multivariate` - upper bound of multivariate confidence interval
- `lower_multivariate` - lower bound of multivariate confidence interval

Examples

```
# Observed Scores
x <- c(
  vci = 130,
  vsi = 130,
  fri = 70,
  wmi = 130,
  psi = 130
)

# Reliability Coefficients
r_xx <- c(
  vci = .92,
  vsi = .92,
  fri = .93,
  wmi = .92,
  psi = .88
  )

# Correlation matrix
R <- ("
  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()

 # Covariance matrix
 sigma <- R * 15 ^ 2

 # Population means#'
 mu <- rep(100, 5)

 mci <- multivariate_ci(
   x = x,
   r_xx = r_xx,
   mu = mu,
   sigma = sigma
 )

 mci

 # Conditional covariance of true score estimates
 attr(mci, "conditional_covariance")
```

p_true_score_less_than_threshold

Given a true score, what is the probability a true score is below a threshold?

Description

Assumes multivaraiate normality of true and observed scores

Usage

```
p_true_score_less_than_threshold(x, threshold, rxx, mu = 100, sigma = 15)
```

Arguments

x	observed score
threshold	threshold score
rxx	reliability coefficient (must be between 0 and 1, exclusively)
mu	population mean (default = 100)
sigma	population standard deviation (default = 15)

Value

a probability

Examples

```
# What is the probability that a true score is 70 or less when
# the observed score is 65, the reliability coefficient is .95,
# the population mean is 100, and the population standard
# deviation is 15?
p_true_score_less_than_threshold(
   x = 65,
   threshold = 70,
   rxx = .95,
   mu = 100,
   sigma = 15)
```

rpi *Convert ability (in W scores by default) to relative proficiency index*

Description

Convert ability (in W scores by default) to relative proficiency index

Usage

```
rpi(
  x,
  mu = 500,
  scale = 20/log(9),
  criterion = 0.9,
  reverse = FALSE,
  interpretation = FALSE
)
```

Arguments

`x`	numeric vector of ability scores
`mu`	numeric vector of ability scores of reference group
`scale`	number vector of scaling factor. The default value (`log(9) / 20`) assumes that x and mu are W scores.
`criterion`	numeric proficiency criterion (between 0 and 1, exclusive)
`reverse`	boolean. If TRUE, the criterion refers to the proficiency of the person instead of the proficiency of the peer group. In other words, the role of the x and mu are reversed.
`interpretation`	If TRUE, the rpi's print method will provide an interpretation of the relative proficiency.

Value

numeric

Examples

```
# What is the probability a person with a W score of 540 can pass
# an item that a person with a 500 W score can pass with a
# probability of .90?
rpi(x = 540, mu = 500, criterion = .9)
# Same as above but with an interpretive statement
rpi(x = 540, mu = 500, criterion = .9, interpretation = TRUE)
# When a person with a W score of 540 has a .9 probability of
# passing an item, what is the probability that a person with a W
# score of 500 will pass it?
rpi(x = 540, mu = 500, criterion = .9, reverse = TRUE, interpretation = TRUE)
```

`w2logit` *Convert W scores to logits*

Description

Convert W scores to logits

Usage

```
w2logit(w, refw = 500)
```

Arguments

`w`	numeric vector of W scores
`refw`	numeric vector of reference W scores

Value

numeric vector of logits

Examples

```
w2logit(540)
```

`w2p` *Convert W scores to a probabilities*

Description

Convert W scores to a probabilities

Usage

```
w2p(w = 500, refw = 500)
```

Arguments

`w`	person ability in w-score units
`refw`	item difficulty in w-score units

Value

numeric vector of probabilities

Examples

```
w2p(w = 520, refw = 500)
```

x2standard *Convert x to a standard score*

Description

Convert x to a standard score

Usage

```
x2standard(
  x,
  mu_x = mean(x, na.rm = T),
  sigma_x = stats::sd(x, na.rm = T),
  mu_new = 100,
  sigma_new = 15,
  digits = ifelse(sigma_new == 1, 2, 0)
)
```

Arguments

`x`	a numeric vector
`mu_x`	mean of current scores
`sigma_x`	standard deviation of current scores
`mu_new`	mean of new scores
`sigma_new`	standard deviation of new scores
`digits`	rounding digits

Value

numeric vector

Examples

```
x2standard(13, mu_x = 10, sigma_x = 3)
```

Index