Package 'manCULTA'

July 12, 2025

Title Common and Unique Latent Transition Analysis (CULTA) as a Way to Examine the Trait-State Dynamics of Alcohol Intoxication

Version 0.9.2

Description Research compendium for the manuscript Pesigan, I. J. A., Russell, M. A., Chow, S.-M. (Under Review). Common and Unique Latent Transition Analysis (CULTA) as a Way to Examine the Trait-State Dynamics of Alcohol Intoxication. <doi:10.0000/0000000000>.

URL https://github.com/jeksterslab/manCULTA, https://jeksterslab.github.io/manCULTA/, https://osf.io/gtdmr/, https://doi.org/10.0000/0000000000

BugReports https://github.com/jeksterslab/manCULTA/issues

License GPL (>= 3)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Depends R (>= 4.0.0)

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp

Suggests knitr, rmarkdown, testthat, lavaan

RoxygenNote 7.3.2

NeedsCompilation yes

Author Ivan Jacob Agaloos Pesigan [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-4818-8420>)

Maintainer Ivan Jacob Agaloos Pesigan <r.jeksterslab@gmail.com>

Contents

`AIC.fitculta` *Akaike's Information Criterion*

Description

Akaike's Information Criterion

Usage

```
## S3 method for class 'fitculta'
AIC(object, ...)
```

Arguments

`object`	Object of class `fitculta`.
`...`	additional arguments.

Value

Returns Akaike's information criterion (AIC).

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
```

```
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting ---------------------------------------------------------------
```

```
# NOTE: Model fitting takes time
fit <- FitCULTA2Profiles(data = data)
AIC(fit)

## End(Not run)
```

`anova.fitculta` *Compare Two Nested* `fitculta` *Models Using Scaled Chi-Square Difference Test*

Description

This function compares two `fitculta` models using the Satorra-Bentler scaled chi-square difference test based on log-likelihoods, number of free parameters, and scaling correction factors. It also returns model fit indices for both models, including AIC, BIC, adjusted BIC, and entropy.

Usage

```
## S3 method for class 'fitculta'
anova(object, other, ...)
```

Arguments

`object`	Model object of class `fitculta`.
`other`	Another model object of class `fitculta`.
`...`	additional arguments.

Value

A list with two elements:

`fit` A matrix summarizing model fit indices for both models, including: `logLik`, `df` (number of free parameters), `correction` (scaling factor), `AIC`, `BIC`, `aBIC` (adjusted BIC), and `entropy`.

`diff` A named numeric vector with scaled chi-square difference, degrees of freedom difference, and p-value.

Examples

```
## Not run:
# complete list of R function arguments ------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
```

```
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)
```

```
# data generation ------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting --------------------------------------------------------------
# NOTE: Model fitting takes time
one_profile <- FitCULTA1Profile(data = data)
two_profiles <- FitCULTA2Profiles(data = data)
anova(one_profile, two_profiles)

## End(Not run)
```

`as.data.frame.simculta` *Coerce an Object of Class* `simculta` *to a Data Frame*

Description

Coerce an Object of Class `simculta` to a Data Frame

Usage

```
## S3 method for class 'simculta'
as.data.frame(x, ...)
```

Arguments

x	Object of class simculta.
...	Additional arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
x <- GenCULTA2Profiles(
  n = 10,
  m = 6,
  common_trait_loading = matrix(
    data = c(1, 1.25, 1.50, 1.75),
    ncol = 1
  ),
  common_state_loading = matrix(
    data = c(1, 1.5, 1.75, 2.00),
    ncol = 1
  ),
  mu_t = NULL,
  psi_t = NULL,
  mu_p = NULL,
  psi_p = NULL,
  theta = diag(4),
  mu_profile = cbind(
    c(-3, -3, -3, -3),
    c(3, 3, 3, 3)
  ),
  mu_x = 0,
  sigma_x = 1,
  nu_0 = -3.563,
  kappa_0 = 0.122,
  alpha_0 = -3.586,
  beta_00 = 2.250,
  gamma_00 = 0.063,
  gamma_10 = 0.094,
  phi_0 = 0.311,
  phi_1 = 0,
  psi_s0 = 0.151,
  psi_s = 0.290
)
as.data.frame(x)
```

---

as.matrix.simculta *Coerce an Object of Class* simculta *to a Matrix*

Description

Coerce an Object of Class `simculta` to a Matrix

Usage

```
## S3 method for class 'simculta'
as.matrix(x, ...)
```

Arguments

`x`	Object of class `simculta`.
`...`	Additional arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
x <- GenCULTA2Profiles(
  n = 10,
  m = 6,
  common_trait_loading = matrix(
    data = c(1, 1.25, 1.50, 1.75),
    ncol = 1
  ),
  common_state_loading = matrix(
    data = c(1, 1.5, 1.75, 2.00),
    ncol = 1
  ),
  mu_t = NULL,
  psi_t = NULL,
  mu_p = NULL,
  psi_p = NULL,
  theta = diag(4),
  mu_profile = cbind(
    c(-3, -3, -3, -3),
    c(3, 3, 3, 3)
  ),
  mu_x = 0,
  sigma_x = 1,
  nu_0 = -3.563,
  kappa_0 = 0.122,
  alpha_0 = -3.586,
  beta_00 = 2.250,
  gamma_00 = 0.063,
  gamma_10 = 0.094,
  phi_0 = 0.311,
  phi_1 = 0,
  psi_s0 = 0.151,
  psi_s = 0.290
```

```
)
as.matrix(x)
```

`BIC.fitculta` *Bayesian Information Criterion*

Description

Bayesian Information Criterion

Usage

```
## S3 method for class 'fitculta'
BIC(object, adjust = FALSE, ...)
```

Arguments

`object`	Object of class `fitculta`.
`adjust`	Logical. If `adjust = TRUE`, return the sample size adjusted BIC.
`...`	additional arguments.

Value

Returns Bayesian information criterion (BIC).

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
```

```
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation ------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
```

```
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting ----------------------------------------------------------------
# NOTE: Model fitting takes time
fit <- FitCULTA2Profiles(data = data)
BIC(fit)
BIC(fit, adjust = TRUE)

## End(Not run)
```

`coef.fitculta` *Parameter Estimates*

Description

Parameter Estimates

Usage

```
## S3 method for class 'fitculta'
coef(object, ...)
```

Arguments

`object`	Object of class `fitculta`.
`...`	additional arguments.

Value

Returns a vector of parameter estimates.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
# complete list of R function arguments ------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
```

```
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting ---------------------------------------------------------------
# NOTE: Model fitting takes time
fit <- FitCULTA2Profiles(data = data)
coef(fit)

## End(Not run)
```

confint.fitculta *Confidence Intervals for Parameter Estimates*

Description

Confidence Intervals for Parameter Estimates

Usage

```
## S3 method for class 'fitculta'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

object	Object of class fitculta.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the confidence level required.
...	additional arguments.

Value

Returns a matrix of confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
```

```
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
```

```
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting ----------------------------------------------------------------
# NOTE: Model fitting takes time
fit <- FitCULTA2Profiles(data = data)
confint(fit, level = 0.95)

## End(Not run)
```

---

converged *Convergence Status of a Model Fit*

Description

Checks whether the model fitting procedure for an object of class `fitculta` has successfully converged based on the presence of the string `"THE BEST LOGLIKELIHOOD VALUE HAS BEEN REPLICATED."` in the Mplus output.

Usage

```
converged(object, ...)
```

Arguments

`object`	Object of class `fitculta`.
`...`	additional arguments.

Value

Logical. `TRUE` if the model has converged, `FALSE` otherwise.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
```

```
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting ---------------------------------------------------------
# NOTE: Model fitting takes time
fit <- FitCULTA2Profiles(data = data)
converged(fit)

## End(Not run)
```

`entropy` *Entropy*

Description

Entropy

Usage

```
entropy(object, ...)
```

Arguments

`object`	Object of class `fitculta`.
`...`	additional arguments.

Value

Returns Akaike's information criterion (AIC).

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
```

```
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
```

```
  mu_profile = mu_profile
)

# model fitting ----------------------------------------------------------
# NOTE: Model fitting takes time
fit <- FitCULTA2Profiles(data = data)
entropy(fit)

## End(Not run)
```

`FitCULTA1Profile` *Fit the One-Profile CULTA Model (CUTS Model with AR)*

Description

Fits the one-profile CULTA model using `Mplus`.

Usage

```
FitCULTA1Profile(data, wd = ".", mplus_bin = NULL, starts = 10)
```

Arguments

`data`	R object. Object of class `simculta`.
`wd`	Character string. Working directory.
`mplus_bin`	Character string. Path to Mplus binary. If `mplus_bin = NULL`, the function will try to find the appropriate binary.
`starts`	Positive integer. Number of initial stage starting values.

Value

Returns an object of class `fitculta`. which is a list with the following elements:

- `call`: Function call.
- `fun`: Function used ("FitCULTA1Profile").
- `args`: Function arguments.
- `output`: `Mplus` output files.
- `elapsed`: Elapsed time.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Model Fitting Functions: `FitCULTA2Profiles()`, `FitLTA2Profiles()`, `FitRILTA2Profiles()`, `InputCULTA1Profile()`, `InputCULTA2Profiles()`, `InputLTA2Profiles()`, `InputRILTA2Profiles()`

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
```

```
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting ---------------------------------------------------------
# NOTE: Model fitting takes time
FitCULTA1Profile(data = data)

## End(Not run)
```

`FitCULTA2Profiles` *Fit the Two-Profile CULTA Model*

Description

Fits the two-profile CULTA model using `Mplus`.

Usage

```
FitCULTA2Profiles(
  data,
  wd = ".",
  ncores = 1L,
  mplus_bin = NULL,
  starts = c(20, 4),
  stiterations = 10,
  stscale = 5
)
```

Arguments

data	R object. Object of class simculta.
wd	Character string. Working directory.
ncores	Positive integer. Number of cores to use.
mplus_bin	Character string. Path to Mplus binary. If mplus_bin = NULL, the function will try to find the appropriate binary.
starts	Vector of positive integer of length two. Number of initial stage starts and number of final stage optimizations.
stiterations	Positive integer. Number of initial stage iterations.
stscale	Positive integer. Random start scale.

Value

Returns an object of class fitculta. which is a list with the following elements:

- call: Function call.
- fun: Function used ("FitCULTA2Profiles").
- args: Function arguments.
- output: Mplus output files.
- elapsed: Elapsed time.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Model Fitting Functions: FitCULTA1Profile(), FitLTA2Profiles(), FitRILTA2Profiles(), InputCULTA1Profile(), InputCULTA2Profiles(), InputLTA2Profiles(), InputRILTA2Profiles()

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
```

```
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation ---------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting -----------------------------------------------------------
# NOTE: Model fitting takes time
FitCULTA2Profiles(data = data)

## End(Not run)
```

`FitLTA2Profiles` *Fit the Two-Profile LTA Model*

Description

Fits the two-profile LTA model using `Mplus`.

Usage

```
FitLTA2Profiles(
  data,
  wd = ".",
  ncores = 1L,
  mplus_bin = NULL,
  starts = c(20, 4),
  stiterations = 10,
  stscale = 5
)
```

Arguments

data	R object. Object of class simculta.
wd	Character string. Working directory.
ncores	Positive integer. Number of cores to use.
mplus_bin	Character string. Path to Mplus binary. If mplus_bin = NULL, the function will try to find the appropriate binary.
starts	Vector of positive integer of length two. Number of initial stage starts and number of final stage optimizations.
stiterations	Positive integer. Number of initial stage iterations.
stscale	Positive integer. Random start scale.

Value

Returns an object of class fitculta. which is a list with the following elements:

- call: Function call.
- fun: Function used ("FitLTA2Profiles").
- args: Function arguments.
- output: Mplus output files.
- elapsed: Elapsed time.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Model Fitting Functions: FitCULTA1Profile(), FitCULTA2Profiles(), FitRILTA2Profiles(), InputCULTA1Profile(), InputCULTA2Profiles(), InputLTA2Profiles(), InputRILTA2Profiles()

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
```

```
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation ------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting --------------------------------------------------------------
# NOTE: Model fitting takes time
FitLTA2Profiles(data = data)

## End(Not run)
```

`FitRILTA2Profiles` *Fit the Two-Profile RI-LTA Model*

Description

Fits the two-profile RI-LTA model using `Mplus`.

Usage

```
FitRILTA2Profiles(
  data,
  wd = ".",
  ncores = 1L,
  mplus_bin = NULL,
  starts = c(20, 4),
  stiterations = 10,
  stscale = 5
)
```

Arguments

data	R object. Object of class simculta.
wd	Character string. Working directory.
ncores	Positive integer. Number of cores to use.
mplus_bin	Character string. Path to Mplus binary. If mplus_bin = NULL, the function will try to find the appropriate binary.
starts	Vector of positive integer of length two. Number of initial stage starts and number of final stage optimizations.
stiterations	Positive integer. Number of initial stage iterations.
stscale	Positive integer. Random start scale.

Value

Returns an object of class fitculta. which is a list with the following elements:

- call: Function call.
- fun: Function used ("FitRILTA2Profiles").
- args: Function arguments.
- output: Mplus output files.
- elapsed: Elapsed time.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Model Fitting Functions: FitCULTA1Profile(), FitCULTA2Profiles(), FitLTA2Profiles(), InputCULTA1Profile(), InputCULTA2Profiles(), InputLTA2Profiles(), InputRILTA2Profiles()

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
```

```
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting ---------------------------------------------------------
# NOTE: Model fitting takes time
FitRILTA2Profiles(data = data)

## End(Not run)
```

GenCULTA2Profiles *Simulate Observed Data from a Two-Profile CULTA Model*

Description

Generates data from a two-profile longitudinal CULTA model where profile membership, trait components, and state dynamics are influenced by an observed covariate. The function produces simulated responses incorporating covariate effects, individual differences, and time-varying fluctuations.

Usage

```
GenCULTA2Profiles(
  n,
  m,
  mu_x,
  sigma_x,
  nu_0,
  kappa_0,
  alpha_0,
  beta_00,
  gamma_00,
  gamma_10,
  mu_t,
  psi_t,
  mu_p,
  psi_p,
  common_trait_loading,
  common_state_loading,
  phi_0,
  phi_1,
  psi_s0,
  psi_s,
  theta,
  mu_profile
)
```

Arguments

n	Positive integer. Number of individuals.
m	Positive integer. Number of measurement occasions.
mu_x	Numeric. Mean of the covariate μ_X.
sigma_x	Numeric. Variance of the covariate σ_X.
nu_0	Numeric. Intercept ν_0 for the logistic model of initial profile membership.
kappa_0	Numeric. Covariate effect κ_0 on initial profile membership.
alpha_0	Numeric. Intercept α_0 for the logistic model of profile transitions across time.
beta_00	Numeric. Effect β_{00} for self-persistence in profile 0 transitions.
gamma_00	Numeric. Covariate effect γ_{00} on remaining in profile 0.
gamma_10	Numeric. Covariate effect γ_{10} on transitioning from profile 1 to profile 0.

`mu_t` Numeric or vector of length q. Mean μ_t of the common trait factor. If `mu_t = NULL`, defaults to zero.

`psi_t` Numeric matrix of size $q \times q$. Positive definite covariance matrix Ψ_t for the common trait factor.

`mu_p` Numeric vector of length p. Mean vector $\boldsymbol{\mu}_p$ for unique trait components. If `mu_p = NULL`, defaults to zero.

`psi_p` Numeric matrix of size $p \times p$. Positive definite covariance matrix Ψ_p for unique trait components.

`common_trait_loading` Numeric matrix of size $p \times q$. Factor loading matrix specifying the influence of the common trait on each observed item.

`common_state_loading` Numeric matrix of size $p \times 1$. Factor loading matrix specifying the influence of the common state on each observed item.

`phi_0` Numeric. Autoregressive coefficient ϕ_0 for the common state process in profile 0.

`phi_1` Numeric. Autoregressive coefficient ϕ_1 for the common state process in profile 0.

`psi_s0` Numeric. Variance ψ_{s0} of the initial common state.

`psi_s` Numeric. Innovation variance ψ_s for the common state process.

`theta` Numeric matrix of size $p \times p$. Positive definite covariance matrix $\boldsymbol{\Theta}$ for unique state components.

`mu_profile` Numeric matrix of size $p \times 2$. Profile-specific means for each observed item across two latent profiles.

Details

The `GenCULTA2Profiles()` function generates data for a two-profile CULTA model with a covariate. The CULTA model incorporates a covariate, latent categorical variables, trait components, state components, and profile-specific means to simulate longitudinal data with latent profile transitions.

Let $i \in \{1, \ldots, n\}$ denote the index for individuals, let $t \in \{0, \ldots, m-1\}$ denote the index measurement occasions, let $k \in \{1, \ldots, p\}$ denote the index items, and let $c \in \{0, 1\}$ be the index of the two latent profiles (profile 0 and profile 1). Let q be the trait dimension, $q = 1$ in this context.

Covariate

The covariate is generated from a normal distribution with mean μ_X and variance σ_X.

Latent Categorical Variables

Latent categorical variables represent profile membership for each individual at each measurement occasion. In a two-profile model, profile membership is influenced by a covariate and previous profile status, following a logistic formulation. We distinguish between:

- Initial profile membership (baseline time point)
- Profile transitions across subsequent time points

We describe both components below.

Initial Profile Membership

For the first measurement occasion ($t = 0$), profile membership is determined by the following log-odds for belonging to profile 0 (with profile 1 as the reference category):

$$\begin{pmatrix} \nu_0 + \kappa_0 \times \text{Covariate} & 0 \end{pmatrix}.$$

The corresponding probability of belonging to each profile is given by:

$$\begin{pmatrix} \frac{\exp(\nu_0+\kappa_0\times\text{Covariate})}{\exp(\nu_0+\kappa_0\times\text{Covariate})+1} & \frac{1}{\exp(\nu_0+\kappa_0\times\text{Covariate})+1} \end{pmatrix}.$$

Profile membership at the first occasion is sampled based on these probabilities.

Profile Transitions

For subsequent occasions ($t = 1, \ldots, m-1$), profile transitions depend on the profile at the previous occasion and the covariate. The log-odds for transitioning to profile 0 at time t are given by:

$$\begin{pmatrix} \alpha_0 + \beta_{00} + \gamma_{00} \times \text{Covariate} & 0 \\ \alpha_0 + \gamma_{10} \times \text{Covariate} & 0 \end{pmatrix}.$$

The probability of transitioning to each profile is computed as:

$$\begin{pmatrix} \frac{\exp(\alpha_0+\beta_{00}+\gamma_{00}\times\text{Covariate})}{\exp(\alpha_0+\beta_{00}+\gamma_{00}\times\text{Covariate})+1} & \frac{1}{\exp(\alpha_0+\beta_{00}+\gamma_{00}\times\text{Covariate})+1} \\ \frac{\exp(\alpha_0+\gamma_{10}\times\text{Covariate})}{\exp(\alpha_0+\gamma_{10}\times\text{Covariate})+1} & \frac{1}{\exp(\alpha_0+\gamma_{10}\times\text{Covariate})+1} \end{pmatrix}.$$

Profile membership for each subsequent time point is sampled using these transition probabilities, based on the individual's covariate value and previous profile.

Trait Components

The trait variate captures between-person differences and is composed of a shared (common) component and item-specific (unique) components. The full decomposition is given by:

$$\text{Trait}_i = \text{Common Trait Loading} \times \text{Common Trait}_i + \text{Unique Trait}_i.$$

We describe each component below.

Common Trait

The common trait Common Trait_i represents shared individual differences that influence all items uniformly. It is drawn from a normal distribution with mean μ_t and variance ψ_t:

$$\text{Common Trait}_i \sim \mathcal{N}(\mu_t, \psi_t)$$

The influence of the common trait on each item is determined by the $p \times q$ common trait loading,

Unique Traits

The unique trait component $\text{Unique Trait}_{k,i}$ captures item-specific stable differences and is drawn from a multivariate normal distribution:

$$\text{Unique Trait}_i \sim \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Psi}_{p\times p})$$

Combined Trait Variate

The trait variate for item k and individual i is obtained by combining the common and unique trait components:

$$\text{Trait}_{k,i} = \text{Common Trait Loading}_k \times \text{Common Trait}_i + \text{Unique Trait}_{k,i}.$$

The common trait component introduces shared variance across items, while the unique trait component allows for item-specific differences not explained by the common trait.

State Components

The state variate is composed of two parts: a common state shared across items, and unique states specific to each item. The full decomposition is given by:

$$\text{State}_{k,i,t} = \text{Common State Loading}_k \times \text{Common State}_{i,t} + \text{Unique State}_{k,i,t}.$$

We describe each component below.

Common State

The common state $\text{Common State}_{i,t}$ evolves over time following a first-order autoregressive process:

$$\text{Common State}_{i,t} = \phi_c \times \text{Common State}_{i,t-1} + \zeta_{i,t}.$$

The initial common state is drawn from a normal distribution:

$$\text{Common State}_{i,0} \sim \mathcal{N}\left(0, \psi_{s_0}\right).$$

The innovation term $\zeta_{i,t}$ is normally distributed:

$$\zeta_{i,t} \sim \mathcal{N}\left(0, \psi_s\right).$$

The autoregressive parameter ϕ_c depends on latent profile membership c:

$$\phi_c = \phi_0 + \left(\phi_1 - \phi_0\right) c.$$

Here, ϕ_0 and ϕ_1 represent the autoregressive coefficients for profiles coded as 0 and 1, respectively.

Unique State

The unique state $\text{Unique State}_{k,i,t}$ captures item-specific deviations and is drawn from a multivariate normal distribution:

$$\text{Unique State}_{i,t} \sim \mathcal{N}\left(0, \boldsymbol{\theta}\right)$$

where $\boldsymbol{\theta}$ is the item-level covariance matrix for the unique state component.

Combined State Variate

The state variate for item k, individual i, and time t combines the common and unique state components:

$$\text{State}_{k,i,t} = \text{Common State Loading}_k \times \text{Common State}_{i,t} + \text{Unique State}_{k,i,t}$$

The common state loading parameter $\text{Common State Loading}_k$ controls the influence of the shared state on each item.

Observed Variables

The observed variable is given by

$$Y_{k,i,t} = \mu_{k,c} + \text{Trait}_{k,i} + \text{State}_{k,i,t}$$

where $\mu_{k,c}$ is the profile specific mean, while $\text{Trait}_{k,i}$ and $\text{State}_{k,i,t}$ correspond to the trait and state components of the model.

Value

Returns an object of class `simculta`. which is a list with the following elements:

- `call`: Function call.
- `fun`: Function used ("GenCULTA2Profiles").
- `args`: Function arguments.
- `id`: Vector of ID numbers.
- `covariate`: Vector of covariate values.
- `categorical`: Latent profiles.
- `common_trait`: Common trait.
- `unique_trait`: Unique trait.
- `common_state`: Common state.
- `trait`: Common trait + unique trait.
- `state`: Common state + unique state.
- `data`: Generated data which is a matrix of observed variables generated from the CULTA model with two-profiles.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Data Generation Functions: `GenData()`

Examples

```
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 10 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
```

```
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- diag(1)
mu_t <- 0
psi_p <- diag(p)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 0.151
psi_s <- 0.290
theta <- diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation ------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
```

```
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)
```

GenData *Simulate Data*

Description

The function simulates data using the GenCULTA2Profiles() function.

Usage

```
GenData(taskid)
```

Arguments

taskid	Positive integer. Task ID.

See Also

Other Data Generation Functions: GenCULTA2Profiles()

Examples

```
## Not run:
set.seed(42)
sim <- GenData(taskid = 1)
as.matrix(sim)

## End(Not run)
```

InputCULTA1Profile *Generate Mplus Inout file for the One-Profile CULTA Model (CUTS Model with AR)*

Description

Generates Mplus input file for the one-profile CULTA model.

Usage

```
InputCULTA1Profile(data, wd = ".", starts = 10)
```

Arguments

data	R object. Object of class simculta.
wd	Character string. Working directory.
starts	Positive integer. Number of initial stage starting values.

Value

Writes data and input files in wd.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Model Fitting Functions: FitCULTA1Profile(), FitCULTA2Profiles(), FitLTA2Profiles(), FitRILTA2Profiles(), InputCULTA2Profiles(), InputLTA2Profiles(), InputRILTA2Profiles()

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
```

```
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
```

```
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# generate data and Mplus input files fitting -------------------------------
InputCULTA1Profile(data = data)

## End(Not run)
```

`InputCULTA2Profiles` *Generate Mplus Input file for the Two-Profile CULTA Model*

Description

Generates `Mplus` input file for the two-profile CULTA model.

Usage

```
InputCULTA2Profiles(
  data,
  wd = ".",
  ncores = 1L,
  starts = c(20, 4),
  stiterations = 10,
  stscale = 5
)
```

Arguments

`data`	R object. Object of class `simculta`.
`wd`	Character string. Working directory.
`ncores`	Positive integer. Number of cores to use.
`starts`	Vector of positive integer of length two. Number of initial stage starts and number of final stage optimizations.
`stiterations`	Positive integer. Number of initial stage iterations.
`stscale`	Positive integer. Random start scale.

Value

Writes data and input files in `wd`.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Model Fitting Functions: FitCULTA1Profile(), FitCULTA2Profiles(), FitLTA2Profiles(), FitRILTA2Profiles(), InputCULTA1Profile(), InputLTA2Profiles(), InputRILTA2Profiles()

Examples

```
## Not run:
# complete list of R function arguments -----------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
```

```
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation ------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# generate data and Mplus input files fitting --------------------------------
InputCULTA2Profiles(data = data)

## End(Not run)
```

`InputLTA2Profiles` *Generate Mplus Input file for the Two-Profile LTA Model*

Description

Generates Mplus input file for the two-profile LTA model.

Usage

```
InputLTA2Profiles(
  data,
  wd = ".",
  ncores = 1L,
  starts = c(20, 4),
  stiterations = 10,
  stscale = 5
)
```

Arguments

data	R object. Object of class simculta.
wd	Character string. Working directory.
ncores	Positive integer. Number of cores to use.
starts	Vector of positive integer of length two. Number of initial stage starts and number of final stage optimizations.
stiterations	Positive integer. Number of initial stage iterations.
stscale	Positive integer. Random start scale.

Value

Writes data and input files in wd.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Model Fitting Functions: FitCULTA1Profile(), FitCULTA2Profiles(), FitLTA2Profiles(), FitRILTA2Profiles(), InputCULTA1Profile(), InputCULTA2Profiles(), InputRILTA2Profiles()

Examples

```
## Not run:
# complete list of R function arguments -----------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
```

```
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)
```

```
# data generation ------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# generate data and Mplus input files fitting --------------------------------
InputLTA2Profiles(data = data)

## End(Not run)
```

`InputRILTA2Profiles` *Generate Mplus Input file for the Two-Profile RI-LTA Model*

Description

Generates `Mplus` input file for the two-profile RI-LTA model.

Usage

```
InputRILTA2Profiles(
  data,
  wd = ".",
  ncores = 1L,
  starts = c(20, 4),
  stiterations = 10,
  stscale = 5
)
```

Arguments

data	R object. Object of class simculta.
wd	Character string. Working directory.
ncores	Positive integer. Number of cores to use.
starts	Vector of positive integer of length two. Number of initial stage starts and number of final stage optimizations.
stiterations	Positive integer. Number of initial stage iterations.
stscale	Positive integer. Random start scale.

Value

Writes data and input files in wd.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Model Fitting Functions: FitCULTA1Profile(), FitCULTA2Profiles(), FitLTA2Profiles(), FitRILTA2Profiles(), InputCULTA1Profile(), InputCULTA2Profiles(), InputLTA2Profiles()

Examples

```
## Not run:
# complete list of R function arguments -----------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
```

```
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
```

```
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# generate data and Mplus input files fitting ------------------------------
InputRILTA2Profiles(data = data)

## End(Not run)
```

`logLik.fitculta` *Extract Log-Likelihood*

Description

Extract Log-Likelihood

Usage

```
## S3 method for class 'fitculta'
logLik(object, ...)
```

Arguments

`object`	Object of class `fitculta`.
`...`	additional arguments.

Value

Returns an object of class `logLik`. This is a number with at the attribute, `"df"` (degrees of freedom), giving the number of (estimated) parameters in the model, and `"correction"` which is the scaling correction factor for MLR.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
```

```
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting ---------------------------------------------------------
# NOTE: Model fitting takes time
fit <- FitCULTA2Profiles(data = data)
logLik(fit, level = 0.95)

## End(Not run)
```

`params` *Simulation Parameters*

Description

Simulation Parameters

Usage

```
data(params)
```

Format

A dataframe with 9 rows and 39 columns:

taskid Simulation Task ID.

n Sample size.

separation Level of separation. `0` for moderate, `-1` for low, and `1` for strong.

m Measurement occasions.

mu_x mu_x parameter. Mean of the covariate.

sigma_x $sigma_x$ parameter. Variance of the covariate.

mu_10 mu_{10} parameter. Profile specific mean for profile 0 and item 1.

mu_20 mu_{20} parameter. Profile specific mean for profile 0 and item 2.

mu_30 mu_{30} parameter. Profile specific mean for profile 0 and item 3.

mu_40 mu_{40} parameter. Profile specific mean for profile 0 and item 4.

lambda_t2 $lambda_{t2}$ parameter. Factor loading for the common trait and item 2.

lambda_s2 $lambda_{s2}$ parameter. Factor loading for the common state and item 2.

lambda_t3 $lambda_{t3}$ parameter. Factor loading for the common trait and item 3.

lambda_s3 $lambda_{s3}$ parameter. Factor loading for the common state and item 3.

lambda_t4 $lambda_{t4}$ parameter. Factor loading for the common trait and item 4.

lambda_s4 $lambda_{s4}$ parameter. Factor loading for the common state and item 4.

theta_11 $theta_{11}$ parameter. Unique state variance for item 1.

theta_22 $theta_{22}$ parameter. Unique state variance for item 2.

theta_33 $theta_{33}$ parameter. Unique state variance for item 3.

theta_44 $theta_{44}$ parameter. Unique state variance for item 4.

phi_0 phi_0 parameter. Autoregressive coefficient for profile 0.

psi_t psi_t parameter. Variance in the common trait; reflects stable between-person differences.

psi_p_11 psi_{p11} parameter. Trait-specific item 1 variance.

psi_p_22 psi_{p22} parameter. Trait-specific item 2 variance.

psi_p_33 psi_{p33} parameter. Trait-specific item 3 variance.

psi_p_44 psi_{p44} parameter. Trait-specific item 4 variance.

psi_s0 psi_{s0} parameter. Initial-day variance of the common state; reflects variability in intoxication levels at observation start.

psi_s psi_s parameter. Residual state variance over days; captures within-person daily fluctuations not explained by trait or AR effects.

mu_11 mu_{11} parameter. Profile specific mean for profile 1 and item 1.

mu_21 mu_{21} parameter. Profile specific mean for profile 1 and item 2.

mu_31 mu_{31} parameter. Profile specific mean for profile 1 and item 3.

mu_41 mu_{41} parameter. Profile specific mean for profile 1 and item 4.

phi_1 phi_1 parameter. Autoregressive coefficient for profile 1.

nu_0 nu_0 parameter. Intercept for initial log-odds of profile 0 (vs. profile 1) when $X = 0$.

alpha_0 $alpha_0$ parameter. Baseline log-odds of being in profile 0 across days.

kappa_0 $kappa_0$ parameter. Covariate effect on initial profile membership; higher X increases odds of profile 0.

beta_00 $beta_{00}$ parameter. Increased odds of staying in profile 0 if previously in that profile; reflects persistence.

gamma_00 $gamma_{00}$ parameter. Covariate effect on staying in profile 0; higher X increases persistence.

gamma_10 $gamma_{10}$ parameter. Covariate effect on switching from state to profile 0; higher X increases transition odds.

Author(s)

Ivan Jacob Agaloos Pesigan

---

print.fitculta *Print Method for an Object of Class* fitculta

---

Description

Print Method for an Object of Class `fitculta`

Usage

```
## S3 method for class 'fitculta'
print(x, alpha = NULL, digits = 4, ...)
```

Arguments

`x`	Object of class `fitculta`.
`alpha`	Numeric vector. Significance level α. If `alpha = NULL`, use `alpha = 0.05`.
`digits`	Digits to print.
`...`	additional arguments.

Value

Prints a matrix of standardized regression slopes, standard errors, test statistics, p-values, and confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
```

```
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting ---------------------------------------------------------
# NOTE: Model fitting takes time
fit <- FitCULTA2Profiles(data = data)
print(fit)

## End(Not run)
```

`print.simculta` *Print Method for an Object of Class* `simculta`

Description

Print Method for an Object of Class `simculta`

Usage

```
## S3 method for class 'simculta'
print(x, ...)
```

Arguments

x	Object of class simculta.
...	Additional arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
x <- GenCULTA2Profiles(
  n = 10,
  m = 6,
  common_trait_loading = matrix(
    data = c(1, 1.25, 1.50, 1.75),
    ncol = 1
  ),
  common_state_loading = matrix(
    data = c(1, 1.5, 1.75, 2.00),
    ncol = 1
  ),
  mu_t = NULL,
  psi_t = NULL,
  mu_p = NULL,
  psi_p = NULL,
  theta = diag(4),
  mu_profile = cbind(
    c(-3, -3, -3, -3),
    c(3, 3, 3, 3)
  ),
  mu_x = 0,
  sigma_x = 1,
  nu_0 = -3.563,
  kappa_0 = 0.122,
  alpha_0 = -3.586,
  beta_00 = 2.250,
  gamma_00 = 0.063,
  gamma_10 = 0.094,
  phi_0 = 0.311,
  phi_1 = 0,
  psi_s0 = 0.151,
  psi_s = 0.290
)
print(x)
```

Sim *Simulation Replication*

Description

Simulation Replication

Usage

```
Sim(taskid, repid, output_folder, overwrite, integrity, seed)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
overwrite	Logical. Overwrite existing output in output_folder.
integrity	Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE.
seed	Integer. Random seed.

Value

The output is saved as an external file in output_folder.

Author(s)

Ivan Jacob Agaloos Pesigan

SimFitCULTA2Profiles *Simulation Replication - FitCULTA2Profiles*

Description

Simulation Replication - FitCULTA2Profiles

Usage

```
SimFitCULTA2Profiles(
  taskid,
  repid,
  output_folder,
  seed,
  suffix,
  overwrite,
  integrity
)
```

Arguments

taskid	Positive integer. Task ID.
repid	Positive integer. Replication ID.
output_folder	Character string. Output folder.
seed	Integer. Random seed.
suffix	Character string. Output of manCTMed:::.SimSuffix().
overwrite	Logical. Overwrite existing output in output_folder.
integrity	Logical. If integrity = TRUE, check for the output file integrity when overwrite = FALSE.

Details

This function is executed via the Sim function.

Value

The output is saved as an external file in output_folder.

Author(s)

Ivan Jacob Agaloos Pesigan

SimFN *Simulation File Name*

Description

Simulation File Name

Usage

```
SimFN(output_type, output_folder, suffix)
```

Arguments

output_type	Character string. Output type.
output_folder	Character string. Output folder.
suffix	Character string. Output of manCTMed:::.SimSuffix().

Value

Returns a character string file name with the output_folder in the OS-specific format.

`SimGenData` *Simulation Replication - GenData*

Description

Simulation Replication - GenData

Usage

```
SimGenData(taskid, repid, output_folder, seed, suffix, overwrite, integrity)
```

Arguments

`taskid`	Positive integer. Task ID.
`repid`	Positive integer. Replication ID.
`output_folder`	Character string. Output folder.
`seed`	Integer. Random seed.
`suffix`	Character string. Output of `manCTMed:::.SimSuffix()`.
`overwrite`	Logical. Overwrite existing output in `output_folder`.
`integrity`	Logical. If `integrity = TRUE`, check for the output file integrity when `overwrite = FALSE`.

Details

This function is executed via the `Sim` function.

Value

The output is saved as an external file in `output_folder`.

Author(s)

Ivan Jacob Agaloos Pesigan

SimProj *Simulation Project Name*

Description

Simulation Project Name

Usage

```
SimProj()
```

Value

Returns the project name as a character string.

Author(s)

Ivan Jacob Agaloos Pesigan

summary.fitculta *Summary Method for an Object of Class* fitculta

Description

Summary Method for an Object of Class `fitculta`

Usage

```
## S3 method for class 'fitculta'
summary(object, alpha = NULL, digits = 4, ...)
```

Arguments

`object`	Object of class `fitculta`.
`alpha`	Numeric vector. Significance level α. If `alpha = NULL`, use `alpha = 0.05`.
`digits`	Digits to print.
`...`	additional arguments.

Value

Returns a matrix of standardized regression slopes, standard errors, test statistics, p-values, and confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
```

```
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
  theta = theta,
  mu_profile = mu_profile
)

# model fitting ---------------------------------------------------------
# NOTE: Model fitting takes time
fit <- FitCULTA2Profiles(data = data)
summary(fit)

## End(Not run)
```

`vcov.fitculta` *Sampling Covariance Matrix*

Description

Sampling Covariance Matrix

Usage

```
## S3 method for class 'fitculta'
vcov(object, ...)
```

Arguments

`object`	Object of class `fitculta`.
`...`	additional arguments.

Value

Returns a matrix of the variance-covariance matrix of parameter estimates.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
## Not run:
# complete list of R function arguments -------------------------------------

# random seed for reproducibility
set.seed(42)

# dimensions
n <- 1000 # number of individuals
m <- 6 # measurement occasions
p <- 4 # number of items
q <- 1 # common trait dimension

# covariate parameters
mu_x <- 11.4009
sigma_x <- 24.67566

# profile membership and transition parameters
nu_0 <- -3.563
kappa_0 <- 0.122
alpha_0 <- -3.586
beta_00 <- 2.250
gamma_00 <- 0.063
gamma_10 <- 0.094

# trait parameters
psi_t <- 0.10 * diag(1)
mu_t <- 0
psi_p <- diag(p)
psi_p_1 <- 0.10
psi_p_2 <- 0.10
psi_p_3 <- 0.50
psi_p_4 <- 0.50
```

```
diag(psi_p) <- c(
  psi_p_1,
  psi_p_2,
  psi_p_3,
  psi_p_4
)
mu_p <- rep(x = 0, times = p)
common_trait_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = q
)

# state parameters
common_state_loading <- matrix(
  data = 1,
  nrow = p,
  ncol = 1
)
phi_0 <- 0.000
phi_1 <- 0.311
psi_s0 <- 1.00
psi_s <- 0.25
theta <- 0.15 * diag(p)

# profile-specific means
mu_profile <- cbind(
  c(2.253, 1.493, 1.574, 1.117),
  c(-0.278, -0.165, -0.199, -0.148)
)

# data generation -------------------------------------------------------------
data <- GenCULTA2Profiles(
  n = n,
  m = m,
  mu_x = mu_x,
  sigma_x = sigma_x,
  nu_0 = nu_0,
  kappa_0 = kappa_0,
  alpha_0 = alpha_0,
  beta_00 = beta_00,
  gamma_00 = gamma_00,
  gamma_10 = gamma_10,
  mu_t = mu_t,
  psi_t = psi_t,
  mu_p = mu_p,
  psi_p = psi_p,
  common_trait_loading = common_trait_loading,
  common_state_loading = common_state_loading,
  phi_0 = phi_0,
  phi_1 = phi_1,
  psi_s0 = psi_s0,
  psi_s = psi_s,
```

```
  theta = theta,
  mu_profile = mu_profile
)

# model fitting ------------------------------------------------------------
# NOTE: Model fitting takes time
fit <- FitCULTA2Profiles(data = data)
vcov(fit)

## End(Not run)
```

`WriteData` *Write Data to File*

Description

Generic function to write data to file.

Usage

```
WriteData(x, file, ...)

## S3 method for class 'simculta'
WriteData(x, file, ...)
```

Arguments

`x`	Object of class `simculta`.
`file`	Character string. File name.
`...`	Additional arguments.

Value

Invisibly returns `NULL`. Writes data to file as a side effect.

Methods (by class)

- `WriteData(simculta)`: Method for objects of class `simculta`.

Author(s)

Ivan Jacob Agaloos Pesigan

Index