Package ‘lcsm’

July 24, 2020

Type Package
Title Univariate and Bivariate Latent Change Score Modeling
Date 2020-07-24
Version 0.1.2
Description Helper functions to implement univariate and bivariate latent change score models in R using the 'lavaan' package.
For details about Latent Change Score Modeling (LCSM) see McArdle (2009) <doi:10.1146/annurev.psych.60.110707.163612> and Grimm, An, McArdle, Zonderman and Resnick (2012) <doi:10.1080/10705511.2012.659627>. The package automatically generates 'lavaan' syntax for different model specifications and varying timepoints. The 'lavaan' syntax generated by this package can be returned and further specifications can be added manually. Longitudinal plots as well as simplified path diagrams can be created to visualise data and model specifications. Estimated model parameters and fit statistics can be extracted as data frames. Data for different univariate and bivariate LCSM can be simulated by specifying estimates for model parameters to explore their effects. This package combines the strengths of other R packages like 'lavaan', 'broom', and 'semPlot' by generating 'lavaan' syntax that helps these packages work together.

Depends R (>= 3.5.0)
License GPL-3
Encoding UTF-8
URL https://milanwiedemann.github.io/lcsm/
BugReports https://github.com/milanwiedemann/lcsm/issues
LazyData true
Imports lavaan (>= 0.6.2), dplyr (>= 0.7.4), tibble (>= 1.4.2), magrittr (>= 1.5), rlang (>= 0.1.6), tidyr (>= 0.8.0), ggplot2 (>= 2.2.1), broom (>= 0.5.1), semPlot (>= 1.1), stats (>= 3.5.2), stringr (>= 1.4.0), purrr (>= 0.3.4), data.table, utf8
RoxygenNote 7.1.1
**data_bi_lcsm**

**Suggests** testthat (>= 2.1.0), knitr (>= 1.22), rmarkdown (>= 1.12), shiny (>= 1.4.0)

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Milan Wiedemann [aut, cre] (<https://orcid.org/0000-0003-1991-282X>),
Anke Ehlers [ths] (<https://orcid.org/0000-0002-8742-0192>),
Mental Health Research UK [fnd]

**Maintainer** Milan Wiedemann <milan.wiedemann@gmail.com>

**Repository** CRAN

**Date/Publication** 2020-07-24 18:00:02 UTC

---

**R topics documented:**

- data_bi_lcsm .................................................. 2
- data_uni_lcsm .............................................. 3
- extract_fit .................................................... 4
- extract_param ............................................... 5
- fit_bi_lcsm .................................................... 6
- fit_uni_lcsm ................................................. 8
- plot_lcsm ..................................................... 10
- plot_trajectories .......................................... 12
- rename_lcsm_vars ............................................ 14
- select_bi_cases ............................................. 14
- select_uni_cases ............................................ 15
- sim_bi_lcsm .................................................. 16
- sim_uni_lcsm ................................................ 19
- specify_bi_lcsm ............................................. 21
- specify_uni_lcsm ............................................ 23

---

**Index** 25

---

---

**data_bi_lcsm**

*Longitudinal dataset with repeated measures of two constructs*

---

**Description**

Example dataset with repeated measures of two constructs to illustrate how the package works.

**Usage**

```r
data(data_bi_lcsm)
```
Format

A longitudinal dataset in wide format:

- id: ID variable, unique identifier for each person
- x1: x value at time point 1
- x2: x value at time point 2
- x3: x value at time point 3
- x4: x value at time point 4
- x5: x value at time point 5
- x6: x value at time point 6
- x7: x value at time point 7
- x8: x value at time point 8
- x9: x value at time point 9
- x10: x value at time point 10
- y1: y value at time point 1
- y2: y value at time point 2
- y3: y value at time point 3
- y4: y value at time point 4
- y5: y value at time point 5
- y6: y value at time point 6
- y7: y value at time point 7
- y8: y value at time point 8
- y9: y value at time point 9
- y10: y value at time point 10

Examples

# Load data into global environment
data(data_bi_lcsm)

data_uni_lcsm

Longitudinal dataset with repeated measures of one constructs

Description

Example dataset with repeated measures of one constructs to illustrate how the package works.

Usage

data(data_uni_lcsm)
Format

A longitudinal dataset in wide format:

- id: ID variable, unique identifier for each person
- x1: x value at time point 1
- x2: x value at time point 2
- x3: x value at time point 3
- x4: x value at time point 4
- x5: x value at time point 5
- x6: x value at time point 6
- x7: x value at time point 7
- x8: x value at time point 8
- x9: x value at time point 9
- x10: x value at time point 10

Examples

```r
# Load data into global environment
data(data_uni_lcsm)
```

---

```r
extract_fit

# Extract fit statistics of lavaan objects

extract_fit(..., details = FALSE)
```

**Arguments**

- `...`: lavaan object(s)
- `details`: Logical, if TRUE return all fit statistics. By default this is set to FALSE, a selection (chisq, npar, aic, bic, cfi, rmsea, srmr) of fit statistics is returned.

**Value**

This function returns a tibble.

**References**

extract_param

Examples

```r
# First create a lavaan object
## Not run:
bi_lcsm_01 <- fit_bi_lcsm(data = data_bi_lcsm,
                         var_x = names(data_bi_lcsm)[2:4],
                         var_y = names(data_bi_lcsm)[12:14],
                         model_x = list(alpha_constant = TRUE,
                                         beta = TRUE,
                                         phi = FALSE),
                         model_y = list(alpha_constant = TRUE,
                                         beta = TRUE,
                                         phi = TRUE),
                         coupling = list(delta_lag_xy = TRUE,
                                          xi_lag_yx = TRUE))

# Now extract fit statistics
extract_fit(bi_lcsm_01)
## End(Not run)
```

extract_param

Extract labelled parameters of lavaan objects

Description

Extract labelled parameters of lavaan objects

Usage

```r
extract_param(lavaan_object, printp = FALSE)
```

Arguments

- `lavaan_object` lavaan object.
- `printp` If TRUE convert into easily readable p values.

Value

This function returns a tibble with labelled parameters.

References

David Robinson and Alex Hayes (2019). broom: Convert Statistical Analysis Objects into Tidy Tibbles. R package version 0.5.2. [https://CRAN.R-project.org/package=broom](https://CRAN.R-project.org/package=broom)
Examples

# First create a lavaan object
bi_lcsm_01 <- fit_bi_lcsm(data = data_bi_lcsm,
var_x = names(data_bi_lcsm)[2:4],
var_y = names(data_bi_lcsm)[12:14],
model_x = list(alpha_constant = TRUE,
beta = TRUE,
phi = FALSE),
model_y = list(alpha_constant = TRUE,
beta = TRUE,
phi = TRUE),
coupling = list(delta_lag_xy = TRUE,
xi_lag_yx = TRUE)
)

# Now extract parameter estimates
extract_param(bi_lcsm_01)

fit_bi_lcsm  
Fit bivariate latent change score models

Description

Fit bivariate latent change score models.

Usage

fit_bi_lcsm(
  data,
  var_x,
  var_y,
  model_x,
  model_y,
  coupling,
  mimic = "Mplus",
estimator = "MLR",
missing = "FIML",
return_lavaan_syntax = FALSE,
...     
)

Arguments

data  Wide dataset.
var_x  List of variables measuring one construct of the model.
var_y  List of variables measuring another construct of the model.
model_x List of model specifications (logical) for variables specified in var_x.
• **alpha_constant** (Constant change factor),
• **alpha_piecewise** (Piecewise constant change factors),
• **alpha_piecewise_num** (Changepoint of piecewise constant change factors),
• **alpha_linear** (Linear change factor),
• **beta** (Proportional change factor),
• **phi** (Autoregression of change scores).

**model_y** List of model specifications for variables specified in `var_y`.
• **alpha_constant** (Constant change factor),
• **alpha_piecewise** (Piecewise constant change factors),
• **alpha_piecewise_num** (Changepoint of piecewise constant change factors),
• **alpha_linear** (Linear change factor),
• **beta** (Proportional change factor),
• **phi** (Autoregression of change scores).

**coupling** List of model specifications (logical) for coupling parameters.
• **coupling_piecewise** (Piecewise coupling parameters),
• **coupling_piecewise_num** (Changepoint of piecewise coupling parameters),
• **delta_xy** (True score y predicting subsequent change score x),
• **delta_yx** (True score x predicting subsequent change score y),
• **xi_xy** (Change score y predicting subsequent change score x),
• **xi_yx** (Change score x predicting subsequent change score y).

**mimic** See lavaan.

**estimator** See lavaan.

**missing** See lavaan.

**return_lavaan_syntax**
 Logical, if TRUE return the lavaan syntax used for simulating data. To make it look beautiful use the function cat.

... Additional arguments to be passed to lavaan.

**Value**
This function returns a lavaan class object.

**References**


Examples

```r
# Fit
fit_bi_lcsm(data = data_bi_lcsm,
            var_x = names(data_bi_lcsm)[2:4],
            var_y = names(data_bi_lcsm)[12:14],
            model_x = list(alpha_constant = TRUE,
                            beta = TRUE,
                            phi = FALSE),
            model_y = list(alpha_constant = TRUE,
                            beta = TRUE,
                            phi = TRUE),
            coupling = list(delta_lag_xy = TRUE,
                             xi_lag_yx = TRUE))
```

fit_uni_lcsm

**Fit univariate latent change score models**

Description

Fit univariate latent change score models.

Usage

```r
fit_uni_lcsm(
  data,         # A data frame in "wide" format, i.e. one column for each measurement point and one row for each observation.
  var,          # Vector, specifying the variable names of each measurement point sequentially.
  model,        # List of model specifications (logical) for variables specified in var.
  mimic = "Mplus",  # Estimator
  estimator = "MLR",  # Estimator
  missing = "FIML",  # Estimator
  return_lavaan_syntax = FALSE,  # Return Lavaan syntax
  ...
)
```

Arguments

data

A data frame in "wide" format, i.e. one column for each measurement point and one row for each observation.

var

Vector, specifying the variable names of each measurement point sequentially.

model

List of model specifications (logical) for variables specified in var.
• alpha_constant (Constant change factor),
• alpha_piecewise (Piecewise constant change factors),
• alpha_piecewise_num (Changepoint of piecewise constant change factors),
• alpha_linear (Linear change factor),
• beta (Proportional change factor),
• phi (Autoregression of change scores).

mimic See lavaan.
estimator See lavaan.
missing See lavaan.
return_lavaan_syntax Logical, if TRUE return the lavaan syntax used for simulating data. To make it look beautiful use the function cat.

Value
This function returns a lavaan class object.

References


Examples
# Fit univariate latent change score model
fit_uni_lcsm(data = data_uni_lcsm,
            var = names(data_uni_lcsm)[2:4],
            model = list(alpha_constant = TRUE,
                         beta = FALSE,
                         phi = FALSE))
plot_lcsm

Plot simplified path diagram of univariate and bivariate latent change score models

Description
Plot simplified path diagram of univariate and bivariate latent change score models

Usage
```r
plot_lcsm(
  lavaan_object,
  layout = NULL,
  lavaan_syntax = NULL,
  return_layout_from_lavaan_syntax = FALSE,
  lcsm = c("univariate", "bivariate"),
  lcsm_colours = FALSE,
  curve_covar = 0.5,
  what = "path",
  whatLabels = "est",
  edge.width = 1,
  node.width = 1,
  border.width = 1,
  fixedStyle = 1,
  freeStyle = 1,
  residuals = FALSE,
  label.scale = FALSE,
  sizeMan = 3,
  sizeLat = 5,
  intercepts = FALSE,
  fade = FALSE,
  nCharNodes = 0,
  nCharEdges = 0,
  edge.label.cex = 0.5,
  ...
)
```

Arguments
- **lavaan_object**: lavaan object of a univariate or bivariate latent change score model.
- **layout**: Matrix, specifying number and location of manifest and latent variables of LCS model specified in `lavaan_object`.
- **lavaan_syntax**: String, lavaan syntax of the lavaan object specified in `lavaan_object`. If `lavaan_syntax` is provided a layout matrix will be generated automatically.
- **return_layout_from_lavaan_syntax**: Logical, if TRUE and `lavaan_syntax` is provided, the layout matrix generated for `semPaths` will be returned for inspection of further customisation.
plot_lcsm

lcsm String, specifying whether lavaan_object represent a "univariate" or "bivariate" LCS model.
lcsm_colours Logical, if TRUE the following colours will be used to highlight different parts of the model: Observed variables (White); Latent true scores (Green); Latent change scores (Blue); Change factors (Yellow).
curve_covar See semPaths.
what See semPlot. "path" to show unweighted grey edges, "par" to show parameter estimates as weighted (green/red) edges
whatLabels See semPaths. "label" to show edge names as label, "est" for parameter estimates, "hide" to hide edge labels.
edge.width See semPaths.
node.width See semPaths.
border.width See semPaths.
fixedStyle See semPaths.
freeStyle See semPaths.
residuals See semPaths.
label.scale See semPaths.
sizeMan See semPaths.
sizeLat See semPaths.
intercepts See semPaths.
fade See semPaths.
nCharNodes See semPaths.
nCharEdges See semPaths.
edge.label.cex See semPaths.
...
Other arguments passed on to semPaths.

Value
Plot

References

Examples
lavaan_syntax_uni <- fit_uni_lcsm(data = data_bi_lcsm,
  var = c("x1", "x2", "x3", "x4", "x5"),
  model = list(alpha_constant = TRUE,
               beta = TRUE,
               phi = TRUE),
  return_lavaan_syntax = TRUE,
  return_lavaan_syntax_string = TRUE)
```
lavaan_object_uni <- fit_uni_lcsm(data = data_bi_lcsm,
   var = c("x1", "x2", "x3", "x4", "x5"),
   model = list(alpha_constant = TRUE,
                beta = TRUE,
                phi = TRUE))

plot_lcsm(lavaan_object = lavaan_object_uni,
          what = "cons", whatLabels = "invisible",
          lavaan_syntax = lavaan_syntax_uni,
          lcsm = "univariate")
```

---

**plot_trajectories**

*Plot individual trajectories*

**Description**

Plot individual trajectories

**Usage**

```
plot_trajectories(
  data,
  id_var,
  var_list,
  line_colour = "blue",
  group_var = NULL,
  point_colour = "black",
  line_alpha = 0.2,
  point_alpha = 0.2,
  point_size = 1,
  smooth = FALSE,
  smooth_method = "loess",
  smooth_se = FALSE,
  xlab = "X",
  ylab = "Y",
  scale_x_num = FALSE,
  scale_x_num_start = 1,
  random_sample_frac = 1,
  seed = 1234,
  title_n = FALSE,
  connect_missing = TRUE
)
```
Arguments

- **data**: Dataset in wide format.
- **id_var**: String, specifying id variable.
- **var_list**: Vector, specifying variable names to be plotted in sequential order.
- **line_colour**: String, specifying colour of lines.
- **group_var**: String, specifying variable name of group, each group will get individual colour lines. This overwrites the line_colour argument. Also consider other options to look at trajectories like `facet_wrap` which may be more appropriate.
- **point_colour**: String, specifying colour of points.
- **line_alpha**: Numeric, specifying alpha of lines.
- **point_alpha**: Numeric, specifying alpha of points.
- **point_size**: Numeric, size of point
- **smooth**: Logical, add smoothed conditional means using `geom_smooth`.
- **smooth_method**: String, specifying method to be used for calculating average line, see `geom_smooth`.
- **smooth_se**: Logical, specifying whether to add standard error of average line or not.
- **xlab**: String for x axis label.
- **ylab**: String for y axis label.
- **scale_x_num**: Logical, if TRUE print sequential numbers starting from 1 as x axis labels, if FALSE use variable names.
- **scale_x_num_start**: Numeric, if `scale_x_num = TRUE` this is the starting value of the x axis.
- **random_sample_frac**: The fraction of rows to select (from wide dataset), default is set to 1 (100 percent) of the sample.
- **seed**: Set seed for random sample if `random_sample_frac` argument is used.
- **title_n**: Logical, specifying whether to print title with number and percentage of cases used for the plot.
- **connect_missing**: Logical, specifying whether to connect points by id_var across missing values.

Value

- **ggplot2 object**

Examples

```r
# Create plot for construct x
plot_trajectories(data = data_bi_lcsm,
                   id_var = "id",
                   var_list = c("x1", "x2", "x3", "x4", "x5",
                                 "x6", "x7", "x8", "x9", "x10"))

# Create plot for construct y specifying some other arguments
plot_trajectories(data = data_bi_lcsm,
                   id_var = "id",
                   var_list = c("x1", "x2", "x3", "x4", "x5",
                                 "x6", "x7", "x8", "x9", "x10"),
                   group_var = "group")
```

```r
# Create plot for construct y with some other arguments
plot_trajectories(data = data_bi_lcsm,
                   id_var = "id",
                   var_list = c("x1", "x2", "x3", "x4", "x5",
                                 "x6", "x7", "x8", "x9", "x10"),
                   group_var = "group",
                   smooth = TRUE,
                   smooth_method = "loess")
```
rename_lcsm_vars

Rename variables for univariate and bivariate latent change score models

**Description**

Rename variables for univariate and bivariate latent change score models

**Usage**

```r
rename_lcsm_vars(data, var_x, var_y)
```

**Arguments**

- `data`: Dataset in wide format
- `var_x`: List of variables measuring first construct
- `var_y`: List of variables measuring second construct

**Value**

Dataset in wide format with renamed variables

select_bi_cases

Select cases based on minimum number of available session scores on two longitudinal measures

**Description**

Select cases based on minimum number of available session scores on two longitudinal measures

**Usage**

```r
select_bi_cases(data, id_var, var_list_x, var_list_y, min_count_x, min_count_y)
```
select_uni_cases

Arguments

- **data**: A data frame in "wide" format, i.e. one column for each measurement point and one row for each observation.
- **id_var**: String, specifying id variable.
- **var_list_x**: Vector, specifying variable names of construct X in sequential order.
- **var_list_y**: Vector, specifying variable names of construct Y in sequential order.
- **min_count_x**: Numeric, specifying minimum number of available scores for construct X.
- **min_count_y**: Numeric, specifying minimum number of available scores for construct Y.

Value
tibble

Examples

```r
select_bi_cases(data_bi_lcsm, 
id_var = "id", 
var_list_x = names(data_bi_lcsm)[2:11], 
var_list_y = names(data_bi_lcsm)[12:21], 
min_count_x = 7, 
min_count_y = 7)
```

Select cases based on minimum number of available session scores on one longitudinal measure

Description

Select cases based on minimum number of available session scores on one longitudinal measure

Usage

```r
select_uni_cases(data, id_var, var_list, min_count, return_id_only = FALSE)
```

Arguments

- **data**: Dataset in wide format.
- **id_var**: String, specifying id variable.
- **var_list**: Vector, specifying variable names in sequential order.
- **min_count**: Numeric, specifying minimum number of available scores.
- **return_id_only**: Logical, if TRUE only return ID. This is needed for select_bi_cases

Value
tibble
Examples

```r
select_uni_cases(data_uni_lcsm,
   id_var = "id",
   var_list = names(data_uni_lcsm)[-1],
   min_count = 7)
```

---

**sim_bi_lcsm**  
*Simulate data from bivariate latent change score model parameter estimates*

**Description**

This function simulates data from bivariate latent change score model parameter estimates using `simulateData`.

**Usage**

```r
sim_bi_lcsm(
   timepoints,
   model_x,
   model_x_param = NULL,
   model_y,
   model_y_param = NULL,
   coupling,
   coupling_param = NULL,
   sample.nobs = 500,
   na_x_pct = 0,
   na_y_pct = 0,
   ...
   var_x = "x",
   var_y = "y",
   change_letter_x = "g",
   change_letter_y = "j",
   return_lavaan_syntax = FALSE
)
```

**Arguments**

- **timepoints**: See `specify_bi_lcsm`
- **model_x**: See `specify_bi_lcsm`
- **model_x_param**: List, specifying parameter estimates for the LCS model that has been specified in the argument `model_x`:
  - `gamma_lx1`: Mean of latent true scores x (Intercept),
  - `sigma2_lx1`: Variance of latent true scores x,
  - `sigma2_ux`: Variance of observed scores x,
• alpha_g2: Mean of change factor (g2),
• alpha_g3: Mean of change factor (g3),
• sigma2_g2: Variance of change factor (g2),
• sigma2_g3: Variance of change factor (g3),
• sigma_g2lx1: Covariance of change factor (g2) with the initial true score x (lx1),
• sigma_g3lx1: Covariance of change factor (g3) with the initial true score x (lx1),
• sigma_g2g3: Covariance of change factors (g2 and g3),
• phi_x: Autoregression of change scores x.

model_y
See specify_bi_lcs

model_y_param List, specifying parameter estimates for the LCS model that has been specified in the argument ‘model_y’:
• gamma_ly1: Mean of latent true scores y (Intercept),
• sigma2_ly1: Variance of latent true scores y,
• sigma2_uy: Variance of observed scores y,
• alpha_j2: Mean of change factor (j2),
• alpha_j3: Mean of change factor (j3),
• sigma2_j2: Variance of change factor (j2),
• sigma2_j3: Variance of change factor (j3),
• sigma_j2ly1: Covariance of change factor (j2) with the initial true score x (ly1),
• sigma_j3ly1: Covariance of change factor (j3) with the initial true score x (ly1),
• sigma_j2j3: Covariance of change factors (j2 and j2),
• phi_y: Autoregression of change scores y.

coupling
See specify_bi_lcs

coupling_param List, specifying parameter estimates coupling parameters that have been specified in the argument ‘coupling’:
• sigma_su: Covariance of residuals x and y,
• sigma_ly1lx1: Covariance of intercepts x and y,
• sigma_g2ly1: Covariance of change factor x (g2) with the initial true score y (ly1),
• sigma_g3ly1: Covariance of change factor x (g3) with the initial true score y (ly1),
• sigma_j2lx1: Covariance of change factor y (j2) with the initial true score x (lx1),
• sigma_j3lx1: Covariance of change factor y (j3) with the initial true score x (lx1),
• sigma_j2g2: Covariance of change factors y (j2) and x (g2),
• sigma_j2g3: Covariance of change factors y (j2) and x (g3),
• sigma_j3g2: Covariance of change factors y (j3) and x (g2),
• delta_con_xy: Change score x (t) determined by true score y (t),
- `delta_con_yx`: Change score y (t) determined by true score x (t),
- `delta_lag_xy`: Change score x (t) determined by true score y (t-1),
- `delta_lag_yx`: Change score y (t) determined by true score x (t-1),
- `xi_con_yx`: Change score x (t) determined by change score y (t),
- `xi_con_yx`: Change score y (t) determined by change score x (t),
- `xi_lag_xy`: Change score x (t) determined by change score y (t-1),
- `xi_lag_yx`: Change score y (t) determined by change score x (t-1)

`sample.nobs` Numeric, number of cases to be simulated, see `specify_uni_lcs

`na_x_pct` Numeric, percentage of random missing values in the simulated dataset [0,1]

`na_y_pct` Numeric, percentage of random missing values in the simulated dataset [0,1]

... Arguments to be passed on to `simulateData`

`var_x` See `specify_bi_lcs

`var_y` See `specify_bi_lcs

`change_letter_x` See `specify_bi_lcs

`change_letter_y` See `specify_bi_lcs

`return_lavaan_syntax` Logical, if TRUE return the lavaan syntax used for simulating data. To make it look beautiful use the function `cat`.

**Value**

tibble

**References**


Examples

# Simulate data from bivariate LCS model parameters
sim_bi_lcsm(timepoints = 12,
             na_x_pct = .05,
             na_y_pct = .1,
             model_x = list(alpha_constant = TRUE, beta = TRUE, phi = FALSE),
             model_x_param = list(gamma_lx1 = 21,
                                  sigma2_lx1 = .5,
                                  sigma2_ux = .2,
                                  alpha_g2 = -.4,
                                  sigma2_g2 = .4,
                                  sigma_g2lx1 = .2,
                                  beta_x = -.1),
             model_y = list(alpha_constant = TRUE, beta = TRUE, phi = TRUE),
             model_y_param = list(gamma_ly1 = 5,
                                  sigma2_ly1 = .2,
                                  sigma2_uy = .2,
                                  alpha_j2 = -.2,
                                  sigma2_j2 = .1,
                                  sigma_j2ly1 = .02,
                                  beta_y = -.2,
                                  phi_y = .1),
             coupling = list(delta_lag_xy = TRUE,
                              xi_lag_yx = TRUE),
             coupling_param = list(sigma_su = .01,
                                    sigma_ly1lx1 = .2,
                                    sigma_g2ly1 = .1,
                                    sigma_j2lx1 = .1,
                                    sigma_j2g2 = .01,
                                    delta_lag_xy = .01,
                                    xi_lag_yx = .4),
             return_lavaan_syntax = FALSE)

---

**sim_uni_lcsm**

*Simulate data from univariate latent change score model parameter estimates*

**Description**

This function simulate data from univariate latent change score model parameter estimates using `simulateData`.

**Usage**

```r
sim_uni_lcsm(
  timepoints,
  model,
  model_param = NULL,
  return_lavaan_syntax = FALSE)
```
Arguments

- **timepoints**: See `specify_uni_lcs`m

- **model**: See `specify_uni_lcs`m

- **model_param**: List, specifying parameter estimates for the LCS model that has been specified in the argument `model`
  - `gamma_lx1`: Mean of latent true scores x (Intercept),
  - `sigma2_lx1`: Variance of latent true scores x,
  - `sigma2_ux`: Variance of observed scores x,
  - `alpha_g2`: Mean of change factor (g2),
  - `alpha_g3`: Mean of change factor (g3),
  - `sigma2_g2`: Variance of constant change factor (g2),
  - `sigma2_g3`: Variance of constant change factor (g3),
  - `sigma_g2lx1`: Covariance of constant change factor (g2) with the initial true score x (lx1),
  - `sigma_g3lx1`: Covariance of constant change factor (g3) with the initial true score x (lx1),
  - `sigma_g2g3`: Covariance of change factors (g2 and g2),
  - `phi_x`: Autoregression of change scores x.

- **var**: See `specify_uni_lcs`m

- **change_letter**: See `specify_uni_lcs`m

- **sample.nobs**: Numeric, number of cases to be simulated, see `specify_uni_lcs`m

- **na_pct**: Numeric, percentage of random missing values in the simulated dataset [0,1]

- **...**: Arguments to be passed on to `simulateData`

- **return_lavaan_syntax**: Logical, if TRUE return the lavaan syntax used for simulating data. To make it look beautiful use the function `cat`.

Value

tibble

Examples

# Simulate data from univariate LCS model parameters
sim_uni_lcs(timepoints = 10,
    model = list(alpha_constant = TRUE, beta = FALSE, phi = TRUE),
model_param = list(gamma_lx1 = 21,  
                 sigma2_lx1 = 1.5,  
                 sigma2_ux = .2,  
                 alpha_j2 = -.93,  
                 sigma2_j2 = .1,  
                 sigma_j2lx1 = .2),
return_lavaan_syntax = FALSE,
sample.nobs = 1000,
na_pct = .3)

Description

Specify lavaan model for bivariate latent change score models

Usage

specify_bi_lcsms(timepoints, var_x, model_x, var_y, model_y, coupling,  
                change_letter_x = "g",  
                change_letter_y = "j")

Arguments

timepoints Number of timepoints.
var_x Vector, specifying variables measuring one construct of the model.
model_x List, specifying model specifications (logical) for variables specified in var_x.
  - alphaconstant (Constant change factor),
  - alphapiecewise (Piecewise constant change factors),
  - alphapiecewise_num (Changepoint of piecewise constant change factors),
  - alphalinear (Linear change factor),
  - beta (Proportional change factor),
  - phi (Autoregression of change scores).
var_y Vector, specifying variables measuring another construct of the model.
model_y List, specifying model specifications (logical) for variables specified in var_y.
specify_bi_lcsm

- alpha_constant (Constant change factor),
- alpha_piecewise (Piecewise constant change factors),
- alpha_piecewise_num (Changepoint of piecewise constant change factors),
- alpha_linear (Linear change factor),
- beta (Proportional change factor),
- phi (Autoregression of change scores).

coupling List, specifying coupling parameters.
- coupling_piecewise (Piecewise coupling parameters),
- coupling_piecewise_num (Changepoint of piecewise coupling parameters),
- delta_con_xy (True score y predicting concurrent change score x),
- delta_lag_xy (True score y predicting subsequent change score x),
- delta_con_yx (True score x predicting concurrent change score y),
- delta_lag_yx (True score x predicting subsequent change score y),
- xi_con_xy (Change score y predicting concurrent change score x),
- xi_lag_xy (Change score y predicting subsequent change score x),
- xi_con_yx (Change score x predicting concurrent change score y),
- xi_lag_yx (Change score x predicting subsequent change score y).

c change_letter_x String, specifying letter to be used as change factor for construct x in lavaan syntax.

c change_letter_y String, specifying letter to be used as change factor for construct y in lavaan syntax.

Value

Lavaan model syntax including comments.

References


**Examples**

```r
# Specify bivariate LCS model
lavaan_bi_lcsm_01 <- specify_bi_lcsm(timepoints = 10,
  var_x = "x",
  model_x = list(alpha_constant = TRUE,
    beta = TRUE,
    phi = TRUE),
  var_y = "y",
  model_y = list(alpha_constant = TRUE,
    beta = TRUE,
    phi = TRUE),
  coupling = list(delta_lag_xy = TRUE,
    delta_lag_yx = TRUE),
  change_letter_x = "g",
  change_letter_y = "j")
```

# To look at string simply return the object
lavaan_bi_lcsm_01

# To get a readable output use cat() function
cat(lavaan_bi_lcsm_01)

---

**specify_uni_lcsm**  
*Specify lavaan model for univariate latent change score models*

**Description**

Specify lavaan model for univariate latent change score models

**Usage**

```r
specify_uni_lcsm(timepoints, model, var, change_letter = "g")
```

**Arguments**

- **timepoints**: Number of timepoints.
- **model**: List of model specifications (logical) for the variables specified in variable.
  - alpha_constant: Constant change factor,
  - alpha_piecewise: Piecewise constant change factors,
  - alpha_piecewise_num: Changepoint of piecewise constant change factors,
  - alpha_linear: Linear change factor,
  - beta: Proportional change factor,
  - phi: Autoregression of change scores.
- **var**: String, specifying letter to be used for variables (Usually x or y).
- **change_letter**: String, specifying letter to be used for change factor (Usually g or j).
specify_uni_lcsm

Value
Lavaan model syntax including comments.

References


Examples
# Specify univariate LCS model
lavaan_uni_lcsm_01 <- specify_uni_lcsm(timepoints = 10,
    model = list(alpha_constant = TRUE,
                 beta = TRUE,
                 phi = TRUE),
    var = "x",
    change_letter = "g")

#' # To look at string simply return the object
lavaan_uni_lcsm_01

# To get a readable output use cat() function
cat(lavaan_uni_lcsm_01)
Index

* dataset
  data_bi_lcsm, 2
  data_uni_lcsm, 3

cat, 7, 9, 18, 20

data_bi_lcsm, 2
data_uni_lcsm, 3

extract_fit, 4
extract_param, 5

facet_wrap, 13
fit_bi_lcsm, 6
fit_uni_lcsm, 8

g geom_smooth, 13

lavaan, 7, 9

plot_lcsm, 10
plot_trajectories, 12

rename_lcsm_vars, 14

select_bi_cases, 14
select_uni_cases, 15

semPaths, 10, 11
sim_bi_lcsm, 16
sim_uni_lcsm, 19
simulateData, 16, 18–20
specify_bi_lcsm, 16–18, 21
specify_uni_lcsm, 18, 20, 23