

Package: lcsm (via r-universe)

October 23, 2024

Type Package

Title Univariate and Bivariate Latent Change Score Modelling

Date 2023-02-25

Version 0.3.2

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](https://doi.org/10.1146/annurev.psych.60.110707.163612)> and Grimm, An, McArdle, Zonderman and Resnick (2012) <[doi:10.1080/10705511.2012.659627](https://doi.org/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 MIT + file LICENSE

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), cli

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0), knitr (>= 1.22), rmarkdown (>= 1.12),
shiny (>= 1.4.0)

VignetteBuilder knitr

Config/testthat/edition 3

Repository <https://milanwiedemann.r-universe.dev>

RemoteUrl <https://github.com/milanwiedemann/lscsm>

RemoteRef HEAD

RemoteSha 501d0d242fb54d3ee9b83736972a2dff8164fd65

Contents

data_bi_lscsm	2
data_uni_lscsm	3
extract_fit	4
extract_param	5
fit_bi_lscsm	6
fit_uni_lscsm	8
lscsm_data	10
plot_lscsm	11
plot_trajectories	14
rename_lscsm_vars	16
select_bi_cases	17
select_uni_cases	18
sim_bi_lscsm	18
sim_uni_lscsm	22
specify_bi_lscsm	24
specify_uni_lscsm	26
Index	28

data_bi_lscsm

Longitudinal dataset with repeated measures of two constructs

Description

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

Usage

```
data(data_bi_lscsm)
```

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_lscsm)
```

data_uni_lscsm

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_lscsm)
```

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

```
# Load data into global environment
data(data_uni_lscsm)
```

extract_fit	<i>Extract fit statistics of lavaan objects</i>
-------------	---

Description

Extract fit statistics of lavaan objects

Usage

```
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

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/>.

Examples

```
# First create a lavaan object
## Not run:
bi_lscsm_01 <- fit_bi_lscsm(data = data_bi_lscsm,
  var_x = names(data_bi_lscsm)[2:4],
  var_y = names(data_bi_lscsm)[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_lscsm_01)

## End(Not run)
```

extract_param	<i>Extract labelled parameters of lavaan objects</i>
---------------	--

Description

Extract labelled parameters of lavaan objects

Usage

```
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/>

Examples

```
# First create a lavaan object
bi_lscsm_01 <- fit_bi_lscsm(data = data_bi_lscsm,
  var_x = names(data_bi_lscsm)[2:4],
  var_y = names(data_bi_lscsm)[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_lscsm_01)
```

fit_bi_lscsm

*Fit bivariate latent change score models***Description**

Fit bivariate latent change score models.

Usage

```
fit_bi_lscsm(
  data,
  var_x,
  var_y,
  model_x,
  model_y,
  coupling,
  add = NULL,
  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	<p>List of model specifications (logical) for variables specified in var_x.</p> <ul style="list-style-type: none"> • alpha_constant (Constant change factor), • alpha_piecewise (Piecewise constant change factors), • alpha_piecewise_num (Changepoint of piecewise constant change factors. In an example with 10 repeated measurements, setting alpha_piecewise_num to 5 would estimate two separate constant change factors, a first one for changes up to timepoint 5, and a second one for changes from timepoint 5 onwards (in this example timepoint 10).), • alpha_linear (Linear change factor), • beta (Proportional change factor), • phi (Autoregression of change scores).
model_y	<p>List of model specifications for variables specified in var_y.</p> <ul style="list-style-type: none"> • alpha_constant (Constant change factor), • alpha_piecewise (Piecewise constant change factors), • alpha_piecewise_num (Changepoint of piecewise constant change factors. In an example with 10 repeated measurements, setting alpha_piecewise_num to 5 would estimate two separate constant change factors, a first one for changes up to timepoint 5, and a second one for changes from timepoint 5 onwards (in this example timepoint 10).), • alpha_linear (Linear change factor), • beta (Proportional change factor), • phi (Autoregression of change scores).
coupling	<p>List of model specifications (logical) for coupling parameters.</p> <ul style="list-style-type: none"> • 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).
add	String, lavaan syntax to be added to the model
mimic	See mimic argument in lavOptions .
estimator	See estimator argument in lavOptions .
missing	See missing argument in lavOptions .
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 lavOptions .

Value

This function returns a lavaan class object.

References

- Ghisletta, P., & McArdle, J. J. (2012). Latent Curve Models and Latent Change Score Models Estimated in R. *Structural Equation Modeling: A Multidisciplinary Journal*, 19(4), 651–682. doi:10.1146/annurev.psych.60.110707.163612.
- Grimm, K. J., Ram, N., & Estabrook, R. (2017). *Growth Modeling—Structural Equation and Multilevel Modeling Approaches*. New York: The Guilford Press.
- McArdle, J. J. (2009). Latent variable modeling of differences and changes with longitudinal data. *Annual Review of Psychology*, 60(1), 577–605. doi:10.1146/annurev.psych.60.110707.163612.
- Yves Rosseel (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software*, 48(2), 1-36. doi:10.18637/jss.v048.i02.

Examples

```
# Fit
fit_bi_lscsm(data = data_bi_lscsm,
             var_x = names(data_bi_lscsm)[2:4],
             var_y = names(data_bi_lscsm)[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_lscsm

Fit univariate latent change score models

Description

Fit univariate latent change score models.

Usage

```
fit_uni_lscsm(
  data,
  var,
  model,
  add = NULL,
  mimic = "Mplus",
  estimator = "MLR",
  missing = "FIML",
  return_lavaan_syntax = FALSE,
  ...
)
```


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. <ul style="list-style-type: none"> • alpha_constant (Constant change factor) • alpha_piecewise (Piecewise constant change factors) • alpha_piecewise_num (Changepoint of piecewise constant change factors. In an example with 10 repeated measurements, setting alpha_piecewise_num to 5 would estimate two separate constant change factors, a first one for changes up to timepoint 5, and a second one for changes from timepoint 5 onwards (in this example timepoint 10)., • alpha_linear (Linear change factor) • beta (Proportional change factor) • phi (Autoregression of change scores)
add	String, lavaan syntax to be added to the model
mimic	See mimic argument in lavOptions .
estimator	See estimator argument in lavOptions .
missing	See missing argument in lavOptions .
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 lavOptions .

Value

This function returns a lavaan class object.

References

- Ghisletta, P., & McArdle, J. J. (2012). Latent Curve Models and Latent Change Score Models Estimated in R. *Structural Equation Modeling: A Multidisciplinary Journal*, 19(4), 651–682. doi:10.1080/10705511.2012.713275.
- Grimm, K. J., Ram, N., & Estabrook, R. (2017). *Growth Modeling—Structural Equation and Multilevel Modeling Approaches*. New York: The Guilford Press.
- McArdle, J. J. (2009). Latent variable modeling of differences and changes with longitudinal data. *Annual Review of Psychology*, 60(1), 577–605. doi:10.1146/annurev.psych.60.110707.163612.
- Yves Rosseel (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software*, 48(2), 1-36. doi:10.18637/jss.v048.i02.

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))
```

lcsm_data

Longitudinal dataset with repeated measures of two constructs

Description

Example dataset with 5 repeated measures of two constructs to illustrate how the package works.

Usage

```
data(lcsm_data)
```

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
- 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

Examples

```
# Load data into global environment
data(lcsm_data)
```

plot_lscsm	<i>Plot simplified path diagram of univariate and bivariate latent change score models</i>
------------	--

Description

Note that the following three arguments are needed to create a plot (see below for more details):

- `lavaan_object`: the lavaan fit object needs to be specified together with a
- `lscsm`: a string indicating whether the latent change score model is "univariate" or "bivariate", and
- `lavaan_syntax`: a separate object with the lavaan syntax as a string

Usage

```
plot_lscsm(
  lavaan_object,
  layout = NULL,
  lavaan_syntax = NULL,
  return_layout_from_lavaan_syntax = FALSE,
  lscsm = c("univariate", "bivariate"),
  lscsm_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

<code>lavaan_object</code>	lavaan object of a univariate or bivariate latent change score model.
<code>layout</code>	Matrix, specifying number and location of manifest and latent variables of LCS model specified in <code>lavaan_object</code> .

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 <code>semPaths</code> will be returned for inspection of further customisation.
lscsm	String, specifying whether lavaan_object represent a "univariate" or "bivariate" LCS model.
lscsm_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 <code>semPaths</code> .
what	See <code>semPlot</code> . "path" to show unweighted grey edges, "par" to show parameter estimates as weighted (green/red) edges
whatLabels	See <code>semPaths</code> . "label" to show edge names as label, "est" for parameter estimates, "hide" to hide edge labels.
edge.width	See <code>semPaths</code> .
node.width	See <code>semPaths</code> .
border.width	See <code>semPaths</code> .
fixedStyle	See <code>semPaths</code> .
freeStyle	See <code>semPaths</code> .
residuals	See <code>semPaths</code> .
label.scale	See <code>semPaths</code> .
sizeMan	See <code>semPaths</code> .
sizeLat	See <code>semPaths</code> .
intercepts	See <code>semPaths</code> .
fade	See <code>semPaths</code> .
nCharNodes	See <code>semPaths</code> .
nCharEdges	See <code>semPaths</code> .
edge.label.cex	See <code>semPaths</code> .
...	Other arguments passed on to <code>semPaths</code> .

Value

Plot

References

Sacha Epskamp (2019). `semPlot`: Path Diagrams and Visual Analysis of Various SEM Packages' Output. R package version 1.1.1. <https://CRAN.R-project.org/package=semPlot/>

Examples

```

# Simplified plot of univariate lscsm
lavaan_syntax_uni <- fit_uni_lscsm(
  data = data_bi_lscsm,
  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_lscsm(
  data = data_bi_lscsm,
  var = c("x1", "x2", "x3", "x4", "x5"),
  model = list(
    alpha_constant = TRUE,
    beta = TRUE,
    phi = TRUE
  )
)

plot_lscsm(
  lavaan_object = lavaan_object_uni,
  what = "cons", whatLabels = "invisible",
  lavaan_syntax = lavaan_syntax_uni,
  lscsm = "univariate"
)

## Not run:
# Simplified plot of bivariate lscsm
lavaan_syntax_bi <- fit_bi_lscsm(
  data = data_bi_lscsm,
  var_x = c("x1", "x2", "x3", "x4", "x5"),
  var_y = c("y1", "y2", "y3", "y4", "y5"),
  model_x = list(
    alpha_constant = TRUE,
    beta = TRUE,
    phi = TRUE
  ),
  model_y = list(
    alpha_constant = TRUE,
    beta = TRUE,
    phi = TRUE
  ),
  coupling = list(
    delta_lag_xy = TRUE,
    delta_lag_yx = TRUE
  ),
  return_lavaan_syntax = TRUE,
  return_lavaan_syntax_string = TRUE
)

```

```
)  
  
lavaan_object_bi <- fit_bi_lscsm(  
  data = data_bi_lscsm,  
  var_x = c("x1", "x2", "x3", "x4", "x5"),  
  var_y = c("y1", "y2", "y3", "y4", "y5"),  
  model_x = list(  
    alpha_constant = TRUE,  
    beta = TRUE,  
    phi = TRUE  
  ),  
  model_y = list(  
    alpha_constant = TRUE,  
    beta = TRUE,  
    phi = TRUE  
  ),  
  coupling = list(  
    delta_lag_xy = TRUE,  
    delta_lag_yx = TRUE  
  )  
)  
  
plot_lscsm(  
  lavaan_object = lavaan_object_bi,  
  what = "cons", whatLabels = "invisible",  
  lavaan_syntax = lavaan_syntax_bi,  
  lscsm = "bivariate"  
)  
  
## End(Not run)
```

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

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

Value

ggplot2 object

Examples

```
# Create plot for construct x
plot_trajectories(data = data_bi_lscsm,
                 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_lscsm,
                 id_var = "id",
                 var_list = c("y1", "y2", "y3", "y4", "y5",
                             "y6", "y7", "y8", "y9", "y10"),
                 xlab = "Time", ylab = "Y Score",
                 connect_missing = FALSE, random_sample_frac = 0.5)
```

rename_lscsm_vars	<i>Rename variables for univariate and bivariate latent change score models</i>
-------------------	---

Description

Rename variables for univariate and bivariate latent change score models

Usage

```
rename_lscsm_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	<i>Select cases based on minimum number of available session scores on two longitudinal measures</i>
-----------------	--

Description

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

Usage

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

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

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

select_uni_cases	<i>Select cases based on minimum number of available session scores on one longitudinal measure</i>
------------------	---

Description

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

Usage

```
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

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

sim_bi_lscsm	<i>Simulate data from bivariate latent change score model parameter estimates</i>
--------------	---

Description

This function simulate data from bivariate latent change score model parameter estimates using [simulateData](#).

Usage

```

sim_bi_lscsm(
  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,
  seed = NULL,
  ...,
  var_x = "x",
  var_y = "y",
  change_letter_x = "g",
  change_letter_y = "j",
  return_lavaan_syntax = FALSE
)

```

Arguments

- | | |
|---------------|---|
| timepoints | See specify_bi_lscsm |
| model_x | See specify_bi_lscsm |
| model_x_param | List, specifying parameter estimates for the LCSM that has been specified in the argument 'model_x': <ul style="list-style-type: none"> • gamma_1x1: Mean of latent true scores x (Intercept), • sigma2_1x1: 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_g21x1: Covariance of change factor (g2) with the initial true score x (1x1), • sigma_g31x1: Covariance of change factor (g3) with the initial true score x (1x1), • sigma_g2g3: Covariance of change factors (g2 and g2), • phi_x: Autoregression of change scores x. |
| model_y | See specify_bi_lscsm |
| model_y_param | List, specifying parameter estimates for the LCSM that has been specified in the argument 'model_y': <ul style="list-style-type: none"> • gamma_1y1: Mean of latent true scores y (Intercept), • sigma2_1y1: Variance of latent true scores y, |

	<ul style="list-style-type: none"> • <code>sigma2_uy</code>: Variance of observed scores y, • <code>alpha_j2</code>: Mean of change factor ($j2$), • <code>alpha_j3</code>: Mean of change factor ($j3$), • <code>sigma2_j2</code>: Variance of change factor ($j2$), • <code>sigma2_j3</code>: Variance of change factor ($j3$), • <code>sigma_j2ly1</code>: Covariance of change factor ($j2$) with the initial true score x ($ly1$), • <code>sigma_j3ly1</code>: Covariance of change factor ($j3$) with the initial true score x ($ly1$), • <code>sigma_j2j3</code>: Covariance of change factors ($j2$ and $j3$), • <code>phi_y</code>: Autoregression of change scores y.
<code>coupling</code>	See specify_bi_lscsm
<code>coupling_param</code>	List, specifying parameter estimates coupling parameters that have been specified in the argument 'coupling': <ul style="list-style-type: none"> • <code>sigma_su</code>: Covariance of residuals x and y, • <code>sigma_ly1lx1</code>: Covariance of intercepts x and y, • <code>sigma_g2ly1</code>: Covariance of change factor x ($g2$) with the initial true score y ($ly1$), • <code>sigma_g3ly1</code>: Covariance of change factor x ($g3$) with the initial true score y ($ly1$), • <code>sigma_j2lx1</code>: Covariance of change factor y ($j2$) with the initial true score x ($lx1$), • <code>sigma_j3lx1</code>: Covariance of change factor y ($j3$) with the initial true score x ($lx1$), • <code>sigma_j2g2</code>: Covariance of change factors y ($j2$) and x ($g2$), • <code>sigma_j2g3</code>: Covariance of change factors y ($j2$) and x ($g3$), • <code>sigma_j3g2</code>: Covariance of change factors y ($j3$) and x ($g2$), • <code>delta_con_xy</code>: Change score x (t) determined by true score y (t), • <code>delta_con_yx</code>: Change score y (t) determined by true score x (t), • <code>delta_lag_xy</code>: Change score x (t) determined by true score y ($t-1$), • <code>delta_lag_yx</code>: Change score y (t) determined by true score x ($t-1$), • <code>xi_con_xy</code>: Change score x (t) determined by change score y (t), • <code>xi_con_yx</code>: Change score y (t) determined by change score x (t), • <code>xi_lag_xy</code>: Change score x (t) determined by change score y ($t-1$), • <code>xi_lag_yx</code>: Change score y (t) determined by change score x ($t-1$)
<code>sample.nobs</code>	Numeric, number of cases to be simulated, see specify_uni_lscsm
<code>na_x_pct</code>	Numeric, percentage of random missing values in the simulated dataset (0 to 1)
<code>na_y_pct</code>	Numeric, percentage of random missing values in the simulated dataset (0 to 1)
<code>seed</code>	Set seed for data simulation, see simulateData
<code>...</code>	Arguments to be passed on to simulateData
<code>var_x</code>	See specify_bi_lscsm
<code>var_y</code>	See specify_bi_lscsm

change_letter_x

See [specify_bi_lscsm](#)

change_letter_y

See [specify_bi_lscsm](#)

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

Ghisletta, P., & McArdle, J. J. (2012). Latent Curve Models and Latent Change Score Models Estimated in R. *Structural Equation Modeling: A Multidisciplinary Journal*, 19(4), 651–682. doi:10.1080/10705511.2012.713275.

Grimm, K. J., Ram, N., & Estabrook, R. (2017). *Growth Modeling—Structural Equation and Multilevel Modeling Approaches*. New York: The Guilford Press.

Kievit, R. A., Brandmaier, A. M., Ziegler, G., van Harmelen, A.-L., de Mooij, S. M. M., Moutoussis, M., ... Dolan, R. J. (2018). Developmental cognitive neuroscience using latent change score models: A tutorial and applications. *Developmental Cognitive Neuroscience*, 33, 99–117. doi:10.1016/j.dcn.2017.11.007.

McArdle, J. J. (2009). Latent variable modeling of differences and changes with longitudinal data. *Annual Review of Psychology*, 60(1), 577–605. doi:10.1146/annurev.psych.60.110707.163612.

Yves Rosseel (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software*, 48(2), 1-36. doi:10.18637/jss.v048.i02.

Examples

```
# Simulate data from bivariate LCSM parameters
sim_bi_lscsm(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 = .13,
                     xi_lag_yx = .4),
return_lavaan_syntax = FALSE)

```

sim_uni_lscsm	<i>Simulate data from univariate latent change score model parameter estimates</i>
---------------	--

Description

This function simulate data from univariate latent change score model parameter estimates using [simulateData](#).

Usage

```

sim_uni_lscsm(
  timepoints,
  model,
  model_param = NULL,
  var = "x",
  change_letter = "g",
  sample.nobs = 500,
  na_pct = 0,
  seed = NULL,
  ...,
  return_lavaan_syntax = FALSE
)

```

Arguments

timepoints	See specify_uni_lscsm
model	See specify_uni_lscsm
model_param	List, specifying parameter estimates for the LCSM that has been specified in the argument 'model' <ul style="list-style-type: none"> • gamma_{1x1}: Mean of latent true scores x (Intercept), • sigma_{2_1x1}: Variance of latent true scores x, • sigma_{2_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_g2l1x1: Covariance of constant change factor (g2) with the initial true score x (lx1),
- sigma_g3l1x1: 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_lscsm](#)

change_letter See [specify_uni_lscsm](#)

sample.nobs Numeric, number of cases to be simulated, see [specify_uni_lscsm](#)

na_pct Numeric, percentage of random missing values in the simulated dataset (0 to 1)

seed Set seed for data simulation, see [simulateData](#)

... 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 LCSM parameters
sim_uni_lscsm(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_g2 = -.93,
                                sigma2_g2 = .1,
                                sigma_g2l1x1 = .2,
                                phi_x = .2),
              return_lavaan_syntax = FALSE,
              sample.nobs = 1000,
              na_pct = .3)
```

specify_bi_lscsm

*Specify lavaan model for bivariate latent change score models***Description**

Specify lavaan model for bivariate latent change score models

Usage

```
specify_bi_lscsm(
  timepoints,
  var_x,
  model_x,
  var_y,
  model_y,
  coupling,
  add = NULL,
  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. <ul style="list-style-type: none"> • 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_y	Vector, specifying variables measuring another construct of the model.
model_y	List, specifying model specifications (logical) for variables specified in var_y. <ul style="list-style-type: none"> • 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 = 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_lscsm_01

# To get a readable output use cat() function
cat(lavaan_bi_lscsm_01)

```

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

Description

Specify lavaan model for univariate latent change score models

Usage

```
specify_uni_lscsm(timepoints, var, model, add = NULL, change_letter = "g")
```

Arguments

timepoints	Number of timepoints.
var	String, specifying letter to be used for of variables (Usually x or y).
model	List of model specifications (logical) for the variables specified in variable. <ul style="list-style-type: none"> • 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.
add	String, lavaan syntax to be added to the model
change_letter	String, specifying letter to be used for change factor (Usually g or j).

Value

Lavaan model syntax including comments.

References

Ghisletta, P., & McArdle, J. J. (2012). Latent Curve Models and Latent Change Score Models Estimated in R. *Structural Equation Modeling: A Multidisciplinary Journal*, 19(4), 651–682. doi:10.1080/10705511.2012.713275.

Grimm, K. J., Ram, N., & Estabrook, R. (2017). *Growth Modeling—Structural Equation and Multilevel Modeling Approaches*. New York: The Guilford Press.

McArdle, J. J. (2009). Latent variable modeling of differences and changes with longitudinal data. *Annual Review of Psychology*, 60(1), 577–605. doi:10.1146/annurev.psych.60.110707.163612.

Yves Rosseel (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software*, 48(2), 1-36. doi:10.18637/jss.v048.i02.

Examples

```
# Specify univariate LCSM
lavaan_uni_lscsm_01 <- specify_uni_lscsm(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_lscsm_01

# To get a readable output use cat() function
cat(lavaan_uni_lscsm_01)
```

Index

* dataset

- data_bi_lscsm, 2
- data_uni_lscsm, 3
- lscsm_data, 10

cat, 7, 9, 21, 23

data_bi_lscsm, 2
data_uni_lscsm, 3

extract_fit, 4
extract_param, 5

facet_wrap, 15
fit_bi_lscsm, 6
fit_uni_lscsm, 8

geom_smooth, 15

lavOptions, 7, 9
lscsm_data, 10

plot_lscsm, 11
plot_trajectories, 14

rename_lscsm_vars, 16

select_bi_cases, 17
select_uni_cases, 18
semPaths, 12
sim_bi_lscsm, 18
sim_uni_lscsm, 22
simulateData, 18, 20, 22, 23
specify_bi_lscsm, 19–21, 24
specify_uni_lscsm, 20, 22, 23, 26