

# Package ‘tidyLPA’

February 25, 2026

**Type** Package

**Title** Easily Carry Out Latent Profile Analysis (LPA) Using Open-Source or Commercial Software

**Version** 2.0.2

**Maintainer** Joshua M Rosenberg <jmrosenberg@utk.edu>

**Description** Easily carry out latent profile analysis (‘LPA’), determine the correct number of classes based on best practices, and tabulate and plot the results. Provides functionality to estimate commonly-specified models with free means, variances, and covariances for each profile. Follows a tidy approach, in that output is in the form of a data frame that can subsequently be computed on. Models can be estimated using the free open source ‘R’ packages ‘Mclust’ and ‘OpenMx’, or using the commercial program ‘MPlus’, via the ‘MplusAutomation’ package.

**License** MIT + file LICENSE

**URL** <https://data-edu.github.io/tidyLPA/>

**BugReports** <https://github.com/data-edu/tidyLPA/issues>

**Depends** R (>= 2.10), tidySEM (>= 0.2.10)

**Imports** dplyr, ggplot2 (>= 4.0.1), mclust, methods, MplusAutomation, tibble, rlang

**Suggests** knitr, lme4, missForest, parallel, pillar, rmarkdown, testthat, OpenMx, mix

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.3

**NeedsCompilation** no

**Author** Joshua M Rosenberg [aut, cre],  
Caspar van Lissa [aut],  
Jennifer A Schmidt [ctb],  
Patrick N Beymer [ctb],  
Daniel Anderson [ctb],  
Matthew J. Schell [ctb]

Repository CRAN

Date/Publication 2026-02-25 06:30:08 UTC

## Contents

AHP . . . . .	2
calc_lrt . . . . .	4
compare_solutions . . . . .	5
curry_mac . . . . .	6
empathy . . . . .	7
estimate_profiles . . . . .	8
estimate_profiles_mclust . . . . .	10
estimate_profiles_mplus2 . . . . .	11
estimate_profiles_openmx . . . . .	12
get_data . . . . .	12
get_estimates . . . . .	14
get_fit . . . . .	15
id_edu . . . . .	16
pisaUSA15 . . . . .	17
poms . . . . .	18
print.tidyLPA . . . . .	18
print.tidyProfile . . . . .	19
single_imputation . . . . .	20
tidyLPA . . . . .	22
%>% . . . . .	23
<b>Index</b>	<b>24</b>

---

AHP

*Select best model using analytic hierarchy process*

---

## Description

Integrates information from several fit indices, and selects the best model.

## Usage

```
AHP(
  fitindices,
  relative_importance = c(AIC = 0.2323, AWE = 0.1129, BIC = 0.2525, CLC = 0.0922, KIC =
    0.3101)
)
```

## Arguments

- `fitindices` A matrix or data.frame of fit indices, with colnames corresponding to the indices named in `relative_importance`.
- `relative_importance` A named numeric vector. Names should correspond to columns in `fitindices`, and values represent the relative weight assigned to the corresponding fit index. The default value corresponds to the fit indices and weights assigned by Akogul and Erisoglu. To assign uniform weights (i.e., each index is weighted equally), assign an equal value to all.

## Details

Many fit indices are available for model selection. Following the procedure developed by Akogul and Erisoglu (2017), this function integrates information from several fit indices, and selects the best model, using Saaty's (1990) Analytic Hierarchy Process (AHP). Conceptually, the process consists of the following steps:

1. For each fit index, calculate the amount of support provided for each model, relative to the other models.
2. From these comparisons, obtain a "priority vector" of the amount of support for each model.
3. Compute a weighted average of the priority vectors for all fit indices, with weights based on a simulation study examining each fit index' ability to recover the correct number of clusters (Akogul & Erisoglu, 2016).
4. Select the model with the highest weighted average priority.

## Value

Numeric.

## Author(s)

Caspar J. van Lissa

## Examples

```
iris[,1:4] %>%  
  estimate_profiles(1:4) %>%  
  get_fit() %>%  
  AHP()
```

---

calc_lrt	<i>Lo-Mendell-Rubin likelihood ratio test</i>
----------	---

---

### Description

Implements the ad-hoc adjusted likelihood ratio test (LRT) described in Formula 15 of Lo, Mendell, & Rubin (2001), or LMR LRT.

### Usage

```
calc_lrt(n, null_ll, null_param, null_classes, alt_ll, alt_param, alt_classes)
```

### Arguments

n	Integer. Sample size
null_ll	Numeric. Log-likelihood of the null model.
null_param	Integer. Number of parameters of the null model.
null_classes	Integer. Number of classes of the null model.
alt_ll	Numeric. Log-likelihood of the alternative model.
alt_param	Integer. Number of parameters of the alternative model.
alt_classes	Integer. Number of classes of the alternative model.

### Value

A numeric vector containing the likelihood ratio LR, the ad-hoc corrected LMR, degrees of freedom, and the LMR p-value.

### References

Lo Y, Mendell NR, Rubin DB. Testing the number of components in a normal mixture. *Biometrika*. 2001;88(3):767–778. doi:10.1093/biomet/88.3.767

### Examples

```
calc_lrt(150L, -741.02, 8, 1, -488.91, 13, 2)
```

---

compare_solutions	<i>Compare latent profile models</i>
-------------------	--------------------------------------

---

## Description

Takes an object of class 'tidyLPA', containing multiple latent profile models with different number of classes or model specifications, and helps select the optimal number of classes and model specification.

## Usage

```
compare_solutions(x, statistics = "BIC")
```

## Arguments

x	An object of class 'tidyLPA'.
statistics	Character vector. Which statistics to examine for determining the optimal model. Defaults to 'BIC'.

## Value

An object of class 'bestLPA' and 'list', containing a tibble of fits 'fits', a named vector 'best', indicating which model fit best according to each fit index, a numeric vector 'AHP' indicating the best model according to the [AHP](#), an object 'plot' of class 'ggplot', and a numeric vector 'statistics' corresponding to argument of the same name.

## Author(s)

Caspar J. van Lissa

## Examples

```
iris_subset <- sample(nrow(iris), 20) # so examples execute quickly
results <- iris %>%
  subset(select = c("Sepal.Length", "Sepal.Width",
    "Petal.Length", "Petal.Width")) %>%
  estimate_profiles(1:3) %>%
  compare_solutions()
```

---

 curry\_mac

*Simulated MAC data*


---

### Description

This simulated dataset, based on Curry et al., 2019, contains data on moral relevance and judgment across the seven domains of the Morality As Cooperation scale.

### Usage

```
data(curry_mac)
```

### Format

A data.frame with 1392 rows and 42 variables.

### Details

<b>sex</b>	factor	Self-identified sex of participants, Male, Female, or Transgendered.
<b>age_years</b>	numeric	Participants' age in years.
<b>KinshipR</b>	numeric	Mean score of moral relevance, kinship subscale.
<b>MutualismR</b>	numeric	Mean score of moral relevance, mutualism subscale.
<b>ExchangeR</b>	numeric	Mean score of moral relevance, exchange subscale.
<b>HawkR</b>	numeric	Mean score of moral relevance, hawk subscale.
<b>DoveR</b>	numeric	Mean score of moral relevance, dove subscale.
<b>DivisionR</b>	numeric	Mean score of moral relevance, division subscale.
<b>PossessionR</b>	numeric	Mean score of moral relevance, possession subscale.
<b>KinshipJ</b>	numeric	Mean score of moral judgment, kinship subscale.
<b>MutualismJ</b>	numeric	Mean score of moral judgment, mutualism subscale.
<b>ExchangeJ</b>	numeric	Mean score of moral judgment, exchange subscale.
<b>HawkJ</b>	numeric	Mean score of moral judgment, hawk subscale.
<b>DoveJ</b>	numeric	Mean score of moral judgment, dove subscale.
<b>DivisionJ</b>	numeric	Mean score of moral judgment, division subscale.
<b>PossessionJ</b>	numeric	Mean score of moral judgment, possession subscale.

### References

Curry, O. S., Jones Chesters, M., & Van Lissa, C. J. (2019). Mapping morality with a compass: Testing the theory of 'morality-as-cooperation' with a new questionnaire. *Journal of Research in Personality*, 78, 106–124. doi:10.1016/j.jrp.2018.10.008

empathy

*Simulated empathy data***Description**

This simulated dataset, based on Van Lissa et al., 2014, contains six annual assessments of adolescents' mean scores on the empathic concern and perspective taking subscales of the Interpersonal Reactivity Index (Davis, 1983). The first measurement wave occurred when adolescents were, on average, 13 years old, and the last one when they were 18 years old.

**Usage**

```
data(empathy)
```

**Format**

A data frame with 467 rows and 13 variables.

**Details**

<b>ec1</b>	numeric	Mean score of empathic concern in wave 1
<b>ec2</b>	numeric	Mean score of empathic concern in wave 2
<b>ec3</b>	numeric	Mean score of empathic concern in wave 3
<b>ec4</b>	numeric	Mean score of empathic concern in wave 4
<b>ec5</b>	numeric	Mean score of empathic concern in wave 5
<b>ec6</b>	numeric	Mean score of empathic concern in wave 6
<b>pt1</b>	numeric	Mean score of perspective taking in wave 1
<b>pt2</b>	numeric	Mean score of perspective taking in wave 2
<b>pt3</b>	numeric	Mean score of perspective taking in wave 3
<b>pt4</b>	numeric	Mean score of perspective taking in wave 4
<b>pt5</b>	numeric	Mean score of perspective taking in wave 5
<b>pt6</b>	numeric	Mean score of perspective taking in wave 6
<b>sex</b>	factor	Adolescent sex; M = male, F = female.

**References**

Van Lissa, C. J., Hawk, S. T., Branje, S. J., Koot, H. M., Van Lier, P. A., & Meeus, W. H. (2014). Divergence Between Adolescent and Parental Perceptions of Conflict in Relationship to Adolescent Empathy Development. *Journal of Youth and Adolescence*, (Journal Article), 1–14. doi:10.1007/s1096401401525

---

estimate\_profiles      *Estimate latent profiles*

---

## Description

Estimates latent profiles (finite mixture models) using either the open source package [mclust](#) or [OpenMx:mxModel][OpenMx], or the commercial program Mplus (using the R-interface of [MplusAutomation](#)).

## Usage

```
estimate_profiles(
  df,
  n_profiles,
  models = NULL,
  variances = "equal",
  covariances = "zero",
  package = "mclust",
  select_vars = NULL,
  ...
)
```

## Arguments

df	data.frame of numeric data; continuous indicators are required for mixture modeling.
n_profiles	Integer vector of the number of profiles (or mixture components) to be estimated.
models	Integer vector. Set to NULL by default, and models are constructed from the variances and covariances arguments. See <a href="#">Details</a> for the six models available in tidyLPA.
variances	Character vector. Specifies which variance components to estimate. Defaults to "equal" (constrain variances across profiles); the other option is "varying" (estimate variances freely across profiles). Each element of this vector refers to one of the models you wish to run.
covariances	Character vector. Specifies which covariance components to estimate. Defaults to "zero" (do not estimate covariances; this corresponds to an assumption of conditional independence of the indicators); other options are "equal" (estimate covariances between items, constrained across profiles), and "varying" (free covariances across profiles).
package	Character. Which package to use; 'OpenMx', 'mclust', or 'MplusAutomation' (requires Mplus to be installed). Default: 'OpenMx'.
select_vars	Character. Optional vector of variable names in df, to be used for model estimation. Defaults to NULL, which means all variables in df are used.
...	Additional arguments are passed to the estimating function; i.e., <a href="#">mxRun</a> , <a href="#">Mclust</a> , or <a href="#">mplusModeler</a> .

## Details

Six models are currently available in tidyLPA, corresponding to the most common requirements. All models estimate the observed variable means for each class. The remaining parameters are:

1. Equal variances across classes; no covariances between observed variables
2. Varying variances across classes; no covariances between observed variables
3. Equal variances and equal covariances across classes
4. Varying variances and equal covariances (not available for package = 'mclust')
5. Equal variances and varying covariances (not available for package = 'mclust')
6. Varying variances and varying covariances

Two interfaces are available to estimate these models; specify their numbers in the `models` argument (e.g., `models = 1`, or `models = c(1, 2, 3)`), or specify the variances/covariances to be estimated (e.g.,: `variances = c("equal", "varying")`, `covariances = c("zero", "equal")`). Note that when `package = 'mclust'` is used, `models = c(4, 5)` are not available. Use `package = 'OpenMx'` or `package = 'Mplus'` to estimate these models.

## Value

A list of class 'tidyLPA'.

## Examples

```
# to make example run more quickly
iris_sample <- iris[c(1:10, 51:60, 101:114), ]

# Example 1:
iris_sample %>%
  subset(select = c("Sepal.Length", "Sepal.Width",
    "Petal.Length")) %>%
  estimate_profiles(3)

# Example 2:
iris %>%
  subset(select = c("Sepal.Length", "Sepal.Width",
    "Petal.Length")) %>%
  estimate_profiles(n_profiles = 1:4, models = 1:3)

# Example 3:
iris_sample %>%
  subset(select = c("Sepal.Length", "Sepal.Width",
    "Petal.Length")) %>%
  estimate_profiles(n_profiles = 1:4, variances = c("equal", "varying"),
    covariances = c("zero", "zero"))

if(requireNamespace("OpenMx", quietly = TRUE)){
# to make example run more quickly
```

```
iris_sample <- iris[c(1:10, 51:60, 101:114), ]

# Example 1:
library(OpenMx)
iris_rename <- iris_sample
names(iris_sample) <- letters[seq_along(names(iris_sample))]
iris_sample %>%
  subset(select = c("a", "b", "c")) %>%
  estimate_profiles(3, package = "OpenMx") -> res
}
```

---

estimate\_profiles\_mclust

*Estimate latent profiles using mclust*

---

## Description

Estimates latent profiles (finite mixture models) using the open source package [mclust](#).

## Usage

```
estimate_profiles_mclust(df, n_profiles, model_numbers, select_vars, ...)
```

## Arguments

<code>df</code>	data.frame with two or more columns with continuous variables
<code>n_profiles</code>	Numeric vector. The number of profiles (or mixture components) to be estimated. Each number in the vector corresponds to an analysis with that many mixture components.
<code>model_numbers</code>	Numeric vector. Numbers of the models to be estimated. See <a href="#">estimate_profiles</a> for a description of the models available in tidyLPA.
<code>select_vars</code>	Character. Optional vector of variable names in <code>df</code> , to be used for model estimation. Defaults to NULL, which means all variables in <code>df</code> are used.
<code>...</code>	Parameters passed directly to <a href="#">Mclust</a> . See the documentation of <a href="#">Mclust</a> .

## Value

An object of class 'tidyLPA' and 'list'

## Author(s)

Caspar J. van Lissa

---

`estimate_profiles_mplus2`*Estimate latent profiles using Mplus*

---

## Description

Estimates latent profiles (finite mixture models) using the commercial program Mplus, through the R-interface of [MplusAutomation](#).

## Usage

```
estimate_profiles_mplus2(  
  df,  
  n_profiles,  
  model_numbers,  
  select_vars,  
  ...,  
  keepfiles = FALSE  
)
```

## Arguments

<code>df</code>	data.frame with two or more columns with continuous variables
<code>n_profiles</code>	Numeric vector. The number of profiles (or mixture components) to be estimated. Each number in the vector corresponds to an analysis with that many mixture components.
<code>model_numbers</code>	Numeric vector. Numbers of the models to be estimated. See <a href="#">estimate_profiles</a> for a description of the models available in tidyLPA.
<code>select_vars</code>	Character. Optional vector of variable names in <code>df</code> , to be used for model estimation. Defaults to NULL, which means all variables in <code>df</code> are used.
<code>...</code>	Parameters passed directly to <a href="#">mplusModeler</a> . See the documentation of <a href="#">mplusModeler</a> .
<code>keepfiles</code>	Logical. Whether to retain the files created by <code>mplusModeler</code> (e.g., for future reference, or to manually edit them).

## Value

An object of class 'tidyLPA' and 'list'

## Author(s)

Caspar J. van Lissa

---

`estimate_profiles_openmx`*Estimate latent profiles using OpenMx*

---

**Description**

Estimates latent profiles using the R-package ‘OpenMx’. Note that this requires installing ‘OpenMx’ and running ‘library(OpenMx)’.

**Usage**

```
estimate_profiles_openmx(df, n_profiles, model_numbers, select_vars, ...)
```

**Arguments**

<code>df</code>	data.frame with two or more columns with continuous variables
<code>n_profiles</code>	Numeric vector. The number of profiles (or mixture components) to be estimated. Each number in the vector corresponds to an analysis with that many mixture components.
<code>model_numbers</code>	Numeric vector. Numbers of the models to be estimated. See <a href="#">estimate_profiles</a> for a description of the models available in tidyLPA.
<code>select_vars</code>	Character. Optional vector of variable names in <code>df</code> , to be used for model estimation. Defaults to NULL, which means all variables in <code>df</code> are used.
<code>...</code>	Parameters passed to and from functions.

**Value**

An object of class ‘tidyLPA’ and ‘list’

**Author(s)**

Caspar J. van Lissa

---

`get_data`*Get data from objects generated by tidyLPA*

---

**Description**

Get data from objects generated by tidyLPA.

**Usage**

```
get_data(x, ...)

## S3 method for class 'tidyLPA'
get_data(x, ...)

## S3 method for class 'tidyProfile'
get_data(x, ...)
```

**Arguments**

**x** An object generated by tidyLPA.

**...** further arguments to be passed to or from other methods. They are ignored in this function.

**Value**

If one model is fit, the data is returned in wide format as a tibble. If more than one model is fit, the data is returned in long form. See the examples.

**Methods (by class)**

- `get_data(tidyLPA)`: Get data for a latent profile analysis with multiple numbers of classes and models, of class 'tidyLPA'.
- `get_data(tidyProfile)`: Get data for a single latent profile analysis object, of class 'tidyProfile'.

**Author(s)**

Caspar J. van Lissa

**Examples**

```
## Not run:
if(interactive()){
  library(dplyr)
  # the data is returned in wide form
  results <- iris %>%
    select(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) %>%
    estimate_profiles(3)
  get_data(results)

  # note that if more than one model is fit, the data is returned in long form
  results1 <- iris %>%
    select(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) %>%
    estimate_profiles(c(3, 4))
  get_data(results1)
}

## End(Not run)
```

---

get\_estimates                    *Get estimates from objects generated by tidyLPA*

---

### Description

Get estimates from objects generated by tidyLPA.

### Usage

```
get_estimates(x, ...)  
  
## S3 method for class 'tidyLPA'  
get_estimates(x, ...)  
  
## S3 method for class 'tidyProfile'  
get_estimates(x, ...)
```

### Arguments

x	An object generated by tidyLPA.
...	further arguments to be passed to or from other methods. They are ignored in this function.

### Value

A tibble.

### Methods (by class)

- `get_estimates(tidyLPA)`: Get estimates for a latent profile analysis with multiple numbers of classes and models, of class 'tidyLPA'.
- `get_estimates(tidyProfile)`: Get estimates for a single latent profile analysis object, of class 'tidyProfile'.

### Author(s)

Caspar J. van Lissa

### Examples

```
## Not run:  
if(interactive()){  
  results <- iris %>%  
    select(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) %>%  
    estimate_profiles(3)  
  get_estimates(results)  
  get_estimates(results[[1]])  
}
```

```
## End(Not run)
```

---

get\_fit

*Get fit indices from objects generated by tidyLPA*

---

## Description

Get fit indices from objects generated by tidyLPA.

## Usage

```
get_fit(x, ...)  
  
## S3 method for class 'tidyLPA'  
get_fit(x, ...)  
  
## S3 method for class 'tidyProfile'  
get_fit(x, ...)
```

## Arguments

x                    An object generated by tidyLPA.  
...                   further arguments to be passed to or from other methods. They are ignored in this function.

## Value

A tibble. Learn more at [https://data-edu.github.io/tidyLPA/articles/Introduction\\_to\\_tidyLPA.html#getting-fit-statistics](https://data-edu.github.io/tidyLPA/articles/Introduction_to_tidyLPA.html#getting-fit-statistics)

## Methods (by class)

- `get_fit(tidyLPA)`: Get fit indices for a latent profile analysis with multiple numbers of classes and models, of class 'tidyLPA'.
- `get_fit(tidyProfile)`: Get fit indices for a single latent profile analysis object, of class 'tidyProfile'.

## Author(s)

Caspar J. van Lissa

## Examples

```
## Not run:
if(interactive()){
  results <- iris %>%
    select(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) %>%
    estimate_profiles(3)
  get_fit(results)
  get_fit(results[[1]])
}

## End(Not run)
```

---

 id\_edu

*Simulated identity data*


---

## Description

This simulated dataset, based on Crochetti et al., 2014, contains five annual assessments of adolescents' mean scores on the commitment, exploration (in depth), and reconsideration subscales of the Utrecht-Management of Identity Commitments Scale (Crocetti et al., 2008). The scores reported here reflect the educational identity subscales of this instrument. The first measurement wave occurred when adolescents were, on average, 14 years old, and the last one when they were 18 years old.

## Usage

```
data(id_edu)
```

## Format

A data frame with 443 rows and 16 variables.

## Details

<b>com1</b>	numeric	Mean score of educational commitment in wave 1
<b>exp1</b>	numeric	Mean score of educational exploration in wave 1
<b>rec1</b>	numeric	Mean score of educational reconsideration in wave 1
<b>com2</b>	numeric	Mean score of educational commitment in wave 2
<b>exp2</b>	numeric	Mean score of educational exploration in wave 2
<b>rec2</b>	numeric	Mean score of educational reconsideration in wave 2
<b>com3</b>	numeric	Mean score of educational commitment in wave 3
<b>exp3</b>	numeric	Mean score of educational exploration in wave 3
<b>rec3</b>	numeric	Mean score of educational reconsideration in wave 3
<b>com4</b>	numeric	Mean score of educational commitment in wave 4
<b>exp4</b>	numeric	Mean score of educational exploration in wave 4
<b>rec4</b>	numeric	Mean score of educational reconsideration in wave 4
<b>com5</b>	numeric	Mean score of educational commitment in wave 5

<b>exp5</b>	numeric	Mean score of educational exploration in wave 5
<b>rec5</b>	numeric	Mean score of educational reconsideration in wave 5
<b>sex</b>	factor	Adolescent sex; M = male, F = female.

## References

Crocetti, E., Klimstra, T. A., Hale, W. W., Koot, H. M., & Meeus, W. (2013). Impact of early adolescent externalizing problem behaviors on identity development in middle to late adolescence: A prospective 7-year longitudinal study. *Journal of Youth and Adolescence*, 42(11), 1745-1758. doi:10.1007/s1096401399246

---

pisaUSA15	<i>student questionnaire data with four variables from the 2015 PISA for students in the United States</i>
-----------	--

---

## Description

student questionnaire data with four variables from the 2015 PISA for students in the United States

## Usage

pisaUSA15

## Format

Data frame with columns #'

**broad\_interest** composite measure of students' self reported broad interest

**enjoyment** composite measure of students' self reported enjoyment

**instrumental\_mot** composite measure of students' self reported instrumental motivation

**self\_efficacy** composite measure of students' self reported self efficacy ...

## Source

<http://www.oecd.org/pisa/data/>

---

poms *Apply POMS-coding to data*

---

### Description

Takes in a data.frame, and applies POMS (proportion of of maximum)-coding to the numeric columns.

### Usage

```
poms(data)
```

### Arguments

data            A data.frame.

### Value

A data.frame.

### Author(s)

Caspar J. van Lissa

### Examples

```
data <- data.frame(a = c(1, 2, 2, 4, 1, 6),
                  b = c(6, 6, 3, 5, 3, 4),
                  c = c("a", "b", "b", "t", "f", "g"))
poms(data)
```

---

print.tidyLPA *Print tidyLPA*

---

### Description

S3 method 'print' for class 'tidyLPA'.

### Usage

```
## S3 method for class 'tidyLPA'
print(
  x,
  stats = c("AIC", "BIC", "Entropy", "prob_min", "prob_max", "n_min", "n_max", "BLRT_p"),
  digits = 2,
  na.print = "",
  ...
)
```

**Arguments**

x	An object of class 'tidyLPA'.
stats	Character vector. Statistics to be printed. Default: c("AIC", "BIC", "Entropy", "prob_min", "prob_max", "n_min", "n_max", "BLRT_p" ).
digits	minimal number of significant digits, see <a href="#">print.default</a> .
na.print	a character string which is used to indicate NA values in printed output, or NULL. See <a href="#">print.default</a> .
...	further arguments to be passed to or from other methods. They are ignored in this function.

**Author(s)**

Caspar J. van Lissa

**Examples**

```
## Not run:
if(interactive()){
  iris %>%
    select(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) %>%
    estimate_profiles(3)
}

## End(Not run)
```

---

print.tidyProfile      *Print tidyProfile*

---

**Description**

S3 method 'print' for class 'tidyProfile'.

**Usage**

```
## S3 method for class 'tidyProfile'
print(x, digits = 2, na.print = "", ...)
```

**Arguments**

x	An object of class 'tidyProfile'.
digits	minimal number of significant digits, see <a href="#">print.default</a> .
na.print	a character string which is used to indicate NA values in printed output, or NULL. See <a href="#">print.default</a> .
...	further arguments to be passed to or from other methods. They are ignored in this function.

**Author(s)**

Caspar J. van Lissa

**Examples**

```
## Not run:
if(interactive()){
  tmp <- iris %>%
    select(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) %>%
    estimate_profiles(3)
  tmp[[2]]
}

## End(Not run)
```

---

single_imputation	<i>Apply single imputation to data</i>
-------------------	--

---

**Description**

This function accommodates several methods for single imputation of data. Currently, the following methods are defined:

**imputeData** Applies the mclust native imputation function [imputeData](#).

**missForest** Applies non-parametric, random-forest based data imputation using [missForest](#). Random forests can accommodate any complex interactions and non-linear relations in the data. Simulation studies indicate that this method is preferable to mclust's `imputeData` (see examples).

**Usage**

```
single_imputation(x, method = "imputeData")
```

**Arguments**

x	A data.frame or matrix.
method	Character. Imputation method to apply, Default: 'imputeData'

**Value**

A data.frame

**Author(s)**

Caspar J. van Lissa

**Examples**

```

## Not run:
library(ggplot2)
library(missForest)
library(mclust)

dm <- 2
k <- 3
n <- 100
V <- 4

# Example of one simulation
class <- sample.int(k, n, replace = TRUE)
dat <- matrix(rnorm(n*V, mean = (rep(class, each = V)-1)*dm), nrow = n,
             ncol = V, byrow = TRUE)
results <- estimate_profiles(data.frame(dat), 1:5)
plot_profiles(results)
compare_solutions(results)

# Simulation for parametric data (i.e., all assumptions of latent profile
# analysis met)
simulation <- replicate(100, {
  class <- sample.int(k, n, replace = TRUE)
  dat <- matrix(rnorm(n*V, mean = (rep(class, each = V)-1)*dm), nrow = n,
               ncol = V, byrow = TRUE)

  d <- prodNA(dat)

  d_mf <- missForest(d)$ximp
  m_mf <- Mclust(d_mf, G = 3, "EEI")
  d_im <- imputeData(d, verbose = FALSE)
  m_im <- Mclust(d_im, G = 3, "EEI")

  class_tabl_mf <- sort(prop.table(table(class, m_mf$classification)),
                       decreasing = TRUE)[1:3]
  class_tabl_im <- sort(prop.table(table(class, m_im$classification)),
                       decreasing = TRUE)[1:3]
  c(sum(class_tabl_mf), sum(class_tabl_im))
})
# Performance on average
rowMeans(simulation)
# Performance SD
colSD(t(simulation))
# Plot shows slight advantage for missForest
plotdat <- data.frame(accuracy = as.vector(simulation), model =
                     rep(c("mf", "im"), n))
ggplot(plotdat, aes(x = accuracy, colour = model))+geom_density()

# Simulation for real data (i.e., unknown whether assumptions are met)
simulation <- replicate(100, {
  d <- prodNA(iris[,1:4])

```

```

d_mf <- missForest(d)$ximp
m_mf <- Mclust(d_mf, G = 3, "EEI")
d_im <- imputeData(d, verbose = FALSE)
m_im <- Mclust(d_im, G = 3, "EEI")

class_tabl_mf <- sort(prop.table(table(iris$Species,
                                     m_mf$classification)), decreasing = TRUE)[1:3]
class_tabl_im <- sort(prop.table(table(iris$Species,
                                     m_im$classification)), decreasing = TRUE)[1:3]
c(sum(class_tabl_mf), sum(class_tabl_im))
})

# Performance on average
rowMeans(simulation)
# Performance SD
colSD(t(simulation))
# Plot shows slight advantage for missForest
plotdat <- data.frame(accuracy = as.vector(tmp),
                     model = rep(c("mf", "im"), n))
ggplot(plotdat, aes(x = accuracy, colour = model))+geom_density()

## End(Not run)

```

---

tidyLPA

*tidyLPA: Functionality to carry out Latent Profile Analysis in R*


---

## Description

Latent Profile Analysis (LPA) is a statistical modeling approach for estimating distinct profiles, or groups, of variables. In the social sciences and in educational research, these profiles could represent, for example, how different youth experience dimensions of being engaged (i.e., cognitively, behaviorally, and affectively) at the same time.

## Details

tidyLPA provides the functionality to carry out LPA in R. In particular, tidyLPA provides functionality to specify different models that determine whether and how different parameters (i.e., means, variances, and covariances) are estimated and to specify (and compare solutions for) the number of profiles to estimate.

## Author(s)

**Maintainer:** Joshua M Rosenberg <jmrosenberg@utk.edu>

Authors:

- Caspar van Lissa <c.j.vanlissa@uu.nl>

Other contributors:

- Jennifer A Schmidt <jaschmid@msu.edu> [contributor]

- Patrick N Beymer <beymerpa@msu.edu> [contributor]
- Daniel Anderson <daniela@uoregon.edu> [contributor]
- Matthew J. Schell <schellma@msu.edu> [contributor]

### See Also

Useful links:

- <https://data-edu.github.io/tidyLPA/>
- Report bugs at <https://github.com/data-edu/tidyLPA/issues>

---

%>%

*Pipe*

---

### Description

tidyLPA suggests using the pipe operator, %>%, from the magrittr package (imported here from the dplyr package).

### Arguments

lhs, rhs            An object and a function to apply to it

### Examples

```
# Instead of
subset(iris, select = c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width"))
# you can write
iris %>%
  subset(select = c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width"))
```

# Index

## \* datasets

- curry\_mac, 6
- empathy, 7
- id\_edu, 16
- pisaUSA15, 17

%>%, 23

AHP, 2, 5

calc\_lrt, 4

compare\_solutions, 5

curry\_mac, 6

empathy, 7

estimate\_profiles, 8, 10–12

estimate\_profiles\_mclust, 10

estimate\_profiles\_mplus2, 11

estimate\_profiles\_openmx, 12

get\_data, 12

get\_estimates, 14

get\_fit, 15

id\_edu, 16

imputeData, 20

Mclust, 8, 10

mclust, 8, 10

missForest, 20

MplusAutomation, 8, 11

mplusModeler, 8, 11

mxRun, 8

pisaUSA15, 17

poms, 18

print.default, 19

print.tidyLPA, 18

print.tidyProfile, 19

single\_imputation, 20

tidyLPA, 22

tidyLPA-package (tidyLPA), 22