Package 'latent'

July 19, 2025

Type Package

Title An R package for Latent Variable Modeling

Version 0.1.0

Date 2025-20-01

Description Fit measurement models with discrete or continuous latent variables.

Author Marcos Jiménez [fnd, cre],
Mauricio Garnier-Villarreal [cre],
Vithor R. Franco [cre]

Maintainer Marcos Jiménez <m.j.jimenezhenriquez@vu.nl>, Mauricio Garnier-Villarreal <m.garniervillarreal@vu.nl>, Vithor R. Franco <vithorfranco@gmail.com>

License GPL-3

Imports Rcpp (>= 1.0.11), png, grid, data.table, lavaan

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.2

Encoding UTF-8

URL https://github.com/Marcosjnez/latent, https://marcosjnez.github.io/latent/

BugReports https://github.com/Marcosjnez/latent/issues

NeedsCompilation yes

LazyData TRUE

Year 2025

Contents

`latent-package` *An R package for Latent Variable Modeling*

Description

Fit measurement models with discrete or continuous latent variables.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.
~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Marcos Jiménez [fnd, cre], Mauricio Garnier-Villarreal [cre], Vithor R. Franco [cre]

Maintainer: Marcos Jiménez <m.j.jimenezhenriquez@vu.nl>, Mauricio Garnier-Villarreal <m.garniervillarreal@vu.nl>, Vithor R. Franco <vithorfranco@gmail.com>

References

~~ Literature or other references for background information ~~

See Also

~~ Optional links to other man pages, e.g. ~~ ~~ <pkg> ~~

Examples

```
# simple examples of the most important functions
```

`cfast` *Fit a Confirmatory Factor Analysis (CFA) model with lavaan syntax.*

Description

Fit a Confirmatory Factor Analysis (CFA) model with lavaan syntax.

Usage

```
cfast(data, model = NULL, cor = "pearson", estimator = "ml",
group = NULL, sample.cov = NULL, nobs = NULL,
missing = "pairwise.complete.obs", W = NULL, std.lv = FALSE,
positive = FALSE, do.fit = TRUE, control = NULL)
```

Examples

```
## Not run:
# The famous Holzinger and Swineford (1939) example
HS.model <- ' visual  =~ x1 + x2 + x3
              textual =~ x4 + x5 + x6
              speed   =~ x7 + x8 + x9 '

fit <- cfast(model = HS.model, data = HolzingerSwineford1939)
summary(fit, fit.measures = TRUE)

## End(Not run)
```

`getfit` *Fit indices*

Description

Compute fit indices from any model.

Usage

```
getfit(model)
```

Arguments

`model`	data.frame or matrix of response.

Details

`getfit` computes all the fit indices related to a specific model.

Value

List with the following fit indices:

`AIC`	.
`BIC`	.

References

None yet.

getmodel *Get the default model for Latent Class Analysis.*

Description

Get the default model for Latent Class Analysis.

Usage

```
getmodel(data, model = rep("multinomial", ncol(data)), nclasses = 2L)
```

Arguments

data	data.frame or matrix of response.
item	Character vector with the model for each item.
nclasses	Number of latent classes.
model	List of parameter labels. See 'details' for more information.
constraints	Should the model be checked for identification? Defaults to TRUE.

Details

getmodel generates the model for the probability of belonging to the classes and the conditional response probabilities. These models may be modified by the user to set equality constraints or to fix parameters.

Value

List with the following objects:

none	.
none	.

References

None yet.

lca *Latent Class Analysis.*

Description

Estimate latent class models with gaussian and multinomial item models.

Usage

```
lca(data, item = rep("gaussian", ncol(data)), nclasses = 2L,
model = NULL, control = list(opt = "lbfgs", rstarts = 30L, cores = 1L),
do.fit = TRUE, constraints = TRUE)
```

Arguments

data	data frame or matrix.
nclasses	Number of latent classes.
item	Character vector with the model for each item (i.e., "gaussian" or "multinomial"). Defaults to "gaussian" for all the items.
model	List of parameter labels. See 'details' for more information.
control	List of control parameters for the optimization algorithm. See 'details' for more information.
do.fit	TRUE to fit the model and FALSE to return only the model setup. Defaults to TRUE.
constraints	Should the model be checked for identification? Defaults to TRUE.

Details

lca estimates models with categorical and continuous data.

Value

List with the following objects:

parameters	The model for the logarithm probabilities of the classes.
f	Logarithm likelihood at the maximum.

References

None yet.

---

print.lca *Latent Class Analysis.*

Description

Print the information of latent class models.

Usage

```
lca(data, model = rep("multinomial", ncol(data)), nclasses = 2L,
control = list(opt = "lbfgs", rstarts = 30L, cores = 1L))
```

Arguments

model	Character vector with the model for each item.

Details

None.

Value

Stuff:

df	Degrees of freedom

References

None yet.

`se` *Standard Errors*

Description

Compute standard errors.

Usage

```
se(fit)
```

Arguments

fit	model fitted with lca.
confidence	Coverage of the confidence interval.

Details

Compute standard errors.

Value

List with the following objects:

vcov	Variance-covariance matrix between the parameters.
se	Standard errors.
SE	Standard errors in the model list.

References

None yet.

`simfactor` *Simulate factor structures with misspecification errors.*

Description

Simulate factor and bifactor structures with crossloadings, correlated factors, and more.

Usage

```
simfactor(nfactors = 5, nitems = 6, loadings = "medium",
crossloadings = 0, correlations = 0,
estimator = "minres", fit = "rmsr", misfit = 0,
error_method = "cudeck", efa = FALSE,
ngenerals = 0, loadings_g = "medium", correlations_g = 0,
pure = FALSE,
lambda = NULL, Phi = NULL, Psi = NULL)
```

Arguments

`nfactors`	Number of factors.
`nitems`	Number of items per factor.
`loadings`	Loadings' magnitude on the factors: "low", "medium" or "high". Defaults to "medium".
`crossloadings`	Magnitude of the cross-loadings among the group factors. Defaults to 0.
`correlations_g`	Correlation among the general factors. Defaults to 0.
`correlations`	Correlation among the factors. Defaults to 0.
`estimator`	estimator used to generate population error: "minres" or "ml".
`fit`	Fit index to control the population error.
`misfit`	Misfit value to generate population error.
`error_method`	Method used to control population error: c("yuan", "cudeck"). Defaults to "cudeck".
`efa`	Reproduce the error with EFA or CFA. Defaults to FALSE (CFA).
`ngenerals`	Number of general factors.
`loadings_g`	Loadings' magnitude on the general factors: "low", "medium" or "high". Defaults to "medium".
`pure`	Fix a pure item on each general factor. Defaults to FALSE.
`lambda`	Custom loading matrix. If Phi is NULL, then all the factors will be correlated at the value given in correlations.
`Phi`	Custom Phi matrix. If lambda is NULL, then Phi should be conformable to the loading matrix specified with the above arguments.
`Psi`	Custom Psi matrix.

Details

`simfactor` generates bi-factor and generalized bifactor patterns with cross-loadings, pure items and correlations among the general and group factors. When `crossloading` is different than 0, one cross-loading is introduced for an item pertaining to each group factor. When `pure` is TRUE, one item loading of each group factor is removed so that the item loads entirely on the general factor. To maintain the item communalities constant upon these modifications, the item loading on the other factors may shrunk (if adding cross-loadings) or increase (if setting pure items).

Loading magnitudes may range between 0.3-0.5 ("low"), 0.4-0.6 ("medium") and 0.5-0.7 ("high"). Custom ranges can be supplied as vectors (i.e., c(0.2, 0.5))

Value

List with the following objects:

`lambda`	Population loading matrix.
`Phi`	Population factor correlation matrix.
`Psi`	Population covariance matrix between the errors.
`R`	Model correlation matrix.
`R_error`	Model correlation matrix with misspecification errors.
`uniquenesses`	Population uniquenesses.
`delta`	Minimum of the loss function that correspond to the misfit value.

Index