# Package 'powRICLPM'

October 14, 2022

**Title** Perform Power Analysis for the Random Intercept Cross-Lagged Panel Model

**Version** 0.1.0

**Date** 2022-09-08

**Maintainer** Jeroen Mulder <j.d.mulder@uu.nl>

**Description** Perform user-friendly power analyses for the bivariate random intercept cross-lagged panel model (RI-CLPM). The strategy as proposed by Mulder (2022) <doi:10.1080/10705511.2022.2122467> is implemented. Extended power analysis options include the use of bounded estimation, inclusion of measurement error in the data generating model and estimation model (i.e., the stable trait autoregressive trait state, STARTS, model), imposing various constraints over time on the parameters of the estimation model, among others.

**License** MIT + file LICENSE

**URL** https://jeroendmulder.github.io/powRICLPM/

**BugReports** https://github.com/JeroenDMulder/powRICLPM/issues/

**Depends** R (>= 4.0.0), stats, utils

**Imports** furrr, ggplot2, lavaan (>= 0.6.7), dplyr, progressr, purrr, rlang, future

**Suggests** knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.2.1

**NeedsCompilation** no

**Author** Jeroen Mulder [aut, cre] (<https://orcid.org/0000-0002-5553-0856>), Netherlands Organization for Scientific Research [fnd]

**Repository** CRAN

**Date/Publication** 2022-10-07 16:00:02 UTC

# R topics documented:



---

`give` *Extract information from* `powRICLPM` *object*

---

## Description

Extract information stored within a `powRICLPM` object (internally used by `print.powRICLPM` and `summary.powRICLPM`). See "Details" for which pieces of information can be extracted. The information is presented by condition (i.e., sample size, number of time points, and ICC).

## Usage

```
give(from, what, parameter = NULL)
```

## Arguments

| | |
|---|---|
| `from` | A `powRICLPM` object |
| `what` | A character string denoting the information to extract, either "conditions", "estimation_problems", "results", or "names". |
| `parameter` | (optional) When `what = "results"`, a character string denoting the parameter to extract the results for. |

## Details

The following information can be extracted from the `powRICLPM` object:

- `conditions`: A `data.frame` with the different experimental conditions per row, where each condition is defined by a unique combination of sample size, number of time points and ICC.
- `estimation_problems`: The proportion of fatal errors, inadmissible values, or non-converged estimations (columns) per experimental conditions (row).
- `results`: The average estimate (`Avg`), minimum estimate (`Min`), standard deviation of parameter estimates (`stdDev`), the average standard error (`SEavg`), the mean square error (`MSE`), the average width of the confidence interval (`Acc`), the coverage rate (`Cov`), and the proportion of times the $p$-value was lower than the significance criterion (`Pow`). It requires setting the `parameter = "..."` argument.
- `names`: The parameter names in the condition with the least parameters (i.e., parameter names that apply to each experimental condition).

## Value

A data.frame.

## Examples

```
load(system.file("extdata", "power_preliminary.Rds", package = "powRICLPM"))

# Return data frame with number of estimation problems per experimental condition
give(power_preliminary, "estimation_problems")

# Return data frame with performance measures for "wB2~wA1" per experimental condition
give(power_preliminary, "results", parameter = "wB2~wA1")
```

---

# plot.powRICLPM *Plot results from* powRICLPM *object*

---

## Description

Visualizes (using **ggplot2**) the estimated power across all experimental conditions within the powRICLPM object, *for a specific parameter*. The plots display the relation between sample size (x-axis) and power (y-axis), grouped by the number of time points, and wrapped by the proportion of between-unit variance.

## Usage

```
## S3 method for class 'powRICLPM'
plot(x, y, ..., parameter = NULL)
```

## Arguments

| | |
|---|---|
| x | A powRICLPM object. |
| y | (don't use) |
| ... | (don't use) |
| parameter | Character string of length denoting the parameter to visualize the results for. |

## Details

**Manually creating plots:** plot.powRICLPM() creates arguably the most obvious plot in the context of power analysis: The relation between power and sample size. However, powRICLPM computes several other metrics (e.g., accuracy, coverage rate, etc.) that researchers might want to plot. To this end, I suggest that users use give() to collect other metrics, across all experimental conditions, for a specific parameter, into a data frame first. Then, users can use any plotting function they like to create their own manual plot. See "Details" of give() for a list of measures that are computed by the function.

## Value

A ggplot2 object.

## See Also

give: Extract information (e.g., performance measures) for a specific parameter, across all experimental conditions. This function is used internally in `plot.powRICLPM`.

## Examples

```
load(system.file("extdata", "power_preliminary.Rds", package = "powRICLPM"))

# Visualize power for "wB2~wA1" across all simulation conditions
plot(power_preliminary, parameter = "wB2~wA1")

# Error: No parameter specified
try(plot(power_preliminary))
```

---

`powRICLPM` *Power analysis for the RI-CLPM (and STARTS model)*

---

## Description

Perform a Monte Carlo power analysis for the random intercept cross-lagged panel model (RI-CLPM). This function computes performance metrics (e.g., bias, mean square error, coverage, power, etc) for all RI-CLPM parameters, and can perform power analyses across multiple experimental conditions simultaneously. Conditions are defined in terms of sample size, number of time points, and proportion of between-unit variance (ICC). See "Details" for information on a) the data simulation, b) model estimation, c) internal naming conventions of parameters, d) the option to include measurement errors (i.e., estimating the Stable Trait Autoregressive Trait State model), e) imposing various constraints over time, and f) parallel execution capabilities for speeding up the analysis.

## Usage

```
powRICLPM(
  target_power,
  search_lower = NULL,
  search_upper = NULL,
  search_step = 20,
  sample_size = NULL,
  time_points,
  ICC,
  RI_cor,
  Phi,
  within_cor,
  reliability = 1,
  skewness = 0,
  kurtosis = 0,
  estimate_ME = FALSE,
  alpha = 0.05,
```

```
  reps = 20,
  bootstrap_reps = 1000,
  seed = NA,
  constraints = "none",
  bounds = FALSE,
  estimator = NA,
  save_path = NULL
)
```

## Arguments

| | |
|---|---|
| target_power | A numeric value between 0 and 1, denoting the targeted power level. |
| search_lower | A positive integer, denoting the lower bound of a range of sample sizes. |
| search_upper | A positive integer, denoting the upper bound of a range of sample sizes. |
| search_step | A positive integer, denoting an increment in sample size. |
| sample_size | (optional) An integer (vector), indicating specific sample sizes at which to evaluate power, rather than specifying a range using the search_* arguments. |
| time_points | An integer (vector) with elements at least larger than 3, indicating number of time points. |
| ICC | A double (vector) with elements between 0 and 1, denoting the proportion of (true score) variance at the between-unit level. When measurement error is included in the data generating model, ICC is computed as the variance of the random intercept factor divided by the true score variance (i.e., controlled for measurement error). |
| RI_cor | A double between 0 and 1, denoting the correlation between random intercepts. |
| Phi | A matrix, with standardized autoregressive effects (on the diagonal) and cross-lagged effects (off-diagonal) in the population. Columns represent predictors and rows represent outcomes. |
| within_cor | A double between 0 and 1, denoting the correlation between the within-unit components. |
| reliability | (optional) A numeric value between 0 and 1, denoting the reliability of the variables. |
| skewness | (optional) A numeric value, denoting the skewness values for the observed variables (see simulateData). |
| kurtosis | (optional) A numeric value, denoting the excess kurtosis values (i.e., compared to the kurtosis of a normal distribution) for the observed variables (see simulateData). |
| estimate_ME | (optional) A logical, denoting if measurement error variance should be estimated in the RI-CLPM (see "Details"). |
| alpha | (optional) A double, denoting the significance criterion. |
| reps | A positive integer, denoting the number of Monte Carlo replications to be used during simulations. |
| bootstrap_reps | (optional) A positive integer, denoting the number of bootstrap samples to use for quantifying the uncertainty (i.e., 95% bootstrap confidence interval) around the power analysis results. |

| | |
|---|---|
| seed | An integer of length 1. If multiple cores are used, a seed of length 1 will be used to generate a full L'Ecuyer-CMRG seed for all cores (see furrr_options). |
| constraints | (optional) A character string, specifying the type of constraints that should be imposed on the estimation model (see "Details"). |
| bounds | (optional) A logical, denoting if bounded estimation should be used for the latent variable variances in the model (see "Details"). |
| estimator | (options) A character, denoting the estimator to be used (default: ML, see "Details"). |
| save_path | A character string naming the directory to save (data) files to (used for validation purposes of this package). Variables are saved in alphabetical and numerical order. |

## Details

A rationale for the power analysis strategy implemented in this package can be found in Mulder (2022).

**Data generation:** Data are generated using simulateData from the **lavaan** package. Based on Phi and within_cor, the residual variances and covariances for the within-components at wave 2 and later are computed, such that the within-components themselves have a variance of 1. This implies that the lagged effects in Phi can be interpreted as standardized effects.

**Model estimation:** Data are analyzed using lavaan from the **lavaan** package. The default estimator is maximum likelihood (ML). Other maximum likelihood based estimators implemented in **lavaan** can be specified as well. When skewed or kurtosed data are generated (using the skewness and kurtosis arguments), the estimator defaults to robust maximum likelihood MLR. The population parameter values are used as starting values.

Parameter estimates from non-converged model solutions are discarded from the results. When bounds = FALSE, inadmissible parameter estimates from converged solutions (e.g., a negative random intercept variance) are discarded. When bounds = TRUE, inadmissible parameter estimates are retained following advice by De Jonckere and Rosseel (2022). The results include the minimum estimates for all parameters across replications to diagnose which parameter(s) is (are) the cause of the inadmissible solution.

**Naming conventions for observed and latent variables:** The observed variables in the RI-CLPM are given default names, namely capital letters in alphabetical order, with numbers denoting the measurement occasion. For example, for a bivariate RICLPM with 3 time points, we observe A1, A2, A3, B1, B2, and B3. Their within-components are denoted by wA1, wA2, ..., wB3, respectively. The between-components have RI_ prepended to the variable name, resulting in RI_A and RI_B.

Parameters are denoted using **lavaan** model syntax (see the **lavaan** website). For example, the random intercept variances are denoted by RI_A~~RI_A and RI_B~~RI_B, the cross-lagged effects at the first wave as wB2~wA1 and wA2~wB1, and the autoregressive effects as wA2~wA1 and wB2~wB1. Use give(object, "names") to extract parameter names from the powRICLPM object.

**Measurement errors (STARTS model):** Including measurement error to the RI-CLPM makes the model equivalent to the Stable Trait Autoregressive Trait State (STARTS) model by Kenny and Zautra (2001) without constraints over time. Measurement error can be added to the generated

data through the reliability argument. Setting reliability = 0.8 implies that 80% is true score variance and 20% is measurement error variance; ICC then denotes the proportion of *true score variance* captured by the random intercept factors. Estimating measurement errors (i.e., the STARTS model) is done by setting est_ME = TRUE.

**Imposing constraints:** The following constraints can be imposed on the estimation model using the constraints = "..." argument:

- lagged: Time-invariant autoregressive and cross-lagged effects.
- residuals: Time-invariant residual variances.
- within: Time-invariant lagged effects and residual variances.
- stationarity: Constraints such that at the within-unit level a stationary process is estimated. This included time-invariant lagged effects, and constraints on the residual variances.
- ME: Time-invariant measurement error variances. Only possible when estimate_ME = TRUE.

**Bounded estimation:** Bounded estimation is useful to avoid nonconvergence in small samples (< 100). Here, automatic wide bounds are used as advised by De Jonckere and Rosseel (2022), see optim.bounds in lavOptions. This option can only be used when no constraints are imposed on the estimation model.

**Parallel processing using furrr:** To speed up the analysis, power analysis for multiple experimental conditions can be executed in parallel. This has been implemented using **furrr**. By default the analysis is executed sequentially (i.e., single-core). Parallel execution (i.e., multicore) can be setup using plan, for example plan(multisession, workers = 4). For more information and options, see https://furrr.futureverse.org.

A progress bar displaying the status of the power analysis has been implemented using **progressr**. By default, a simple progress bar will be shown. For more information on how to control this progress bar and several other notification options (e.g., auditory notifications), see https://progressr.futureverse.org.

## Value

A list containing a conditions and session element. condition itself is a list of experimental conditions, where each element is again a list containing the input and output of the power analysis for that particular experimental condition. session is a list containing information common to all experimental conditions.

## Author(s)

Jeroen D. Mulder <j.d.mulder@uu.nl>

## References

De Jonckere, J., & Rosseel, Y. (2022). Using bounded estimation to avoid nonconvergence in small sample structural equation modeling. *Structural Equation Modeling*, *29*(3), 412-427. doi:10.1080/10705511.2021.1982716

Kenny, D. A., & Zautra, A. (2001). Trait–state models for longitudinal data. *New methods for the analysis of change* (pp. 243–263). American Psychological Association. doi:10.1037/10409008

Mulder, J. D. (2022). Power analysis for the random intercept cross-lagged panel model using the *powRICLPM* R-package. *Structural Equation Modeling*. doi:10.1080/10705511.2022.2122467

## See Also

- powRICLPM_Mplus: Create Mplus model syntax for RI-CLPM power analysis.
- summary.powRICLPM: Summarize the setup of powRICLPM object.
- give: Extract information from powRICLPM objects.
- plot.powRICLPM: Visualize results powRICLPM object for a specific parameter.

## Examples

```
# Define population parameters for lagged effects
Phi <- matrix(c(.4, .1, .2, .3), ncol = 2, byrow = TRUE)

# (optional) Set up parallel computing (i.e., multicore, speeding up the analysis)
library(furrr)
library(progressr)
future::plan(multisession)


# Run analysis ("reps" is small, because this is an example)
with_progress({
  power_preliminary <- powRICLPM(
    target_power = 0.8,
    search_lower = 500,
    search_upper = 1000,
    search_step = 100,
    time_points = c(3, 4),
    ICC = c(0.4, 0.5, 0.6),
    RI_cor = 0.3,
    Phi = Phi,
    within_cor = 0.3,
    reps = 100,
    seed = 1234
  )
})
```

---

powRICLPM_Mplus *Create Mplus syntax for RI-CLPM power analysis*

---

## Description

powRICLPM_Mplus() creates and saves (a) syntax file(s) for performing a Monte Carlo power analysis for the random intercept cross-lagged panel model (RI-CLPM) in Mplus. Mplus model syntax can be created across multiple experimental conditions simultaneously. Conditions are defined in terms of sample size, number of time points, and proportion of between-unit variance (ICC). See "Details" for information on a) the naming conventions of parameters, and b) the various constraints that can be imposed on the model.

## Usage

```
powRICLPM_Mplus(
  search_lower = NULL,
  search_upper = NULL,
  search_step = 20,
  sample_size = NULL,
  time_points,
  ICC,
  RI_cor,
  Phi,
  within_cor,
  reps = 1000,
  seed = NA,
  save_path = getwd(),
  constraints = "none"
)
```

## Arguments

| | |
|---|---|
| search_lower | A positive integer, denoting the lower bound of a range of sample sizes. |
| search_upper | A positive integer, denoting the upper bound of a range of sample sizes. |
| search_step | A positive integer, denoting an increment in sample size. |
| sample_size | (optional) An integer (vector), indicating specific sample sizes at which to evaluate power, rather than specifying a range using the search_* arguments. |
| time_points | An integer (vector) with elements at least larger than 3, indicating number of time points. |
| ICC | A double (vector) with elements between 0 and 1, denoting the proportion of (true score) variance at the between-unit level. When measurement error is included in the data generating model, ICC is computed as the variance of the random intercept factor divided by the true score variance (i.e., controlled for measurement error). |
| RI_cor | A double between 0 and 1, denoting the correlation between random intercepts. |
| Phi | A matrix, with standardized autoregressive effects (on the diagonal) and cross-lagged effects (off-diagonal) in the population. Columns represent predictors and rows represent outcomes. |
| within_cor | A double between 0 and 1, denoting the correlation between the within-unit components. |
| reps | A positive integer, denoting the number of Monte Carlo replications to be used during simulations. |
| seed | An integer of length 1. If multiple cores are used, a seed of length 1 will be used to generate a full L'Ecuyer-CMRG seed for all cores (see furrr_options). |
| save_path | A character string, denoting the path to the folder to save the Mplus syntax files in (default: current working directory). |
| constraints | (optional) A character string, specifying the type of constraints that should be imposed on the estimation model (see "Details"). |

## Details

**Syntax generation:** Mplus model syntax is created in multiple steps: First, the `MODEL POPULATION` command syntax is created in which parameters are constrained to population values. Second, the `MODEL` command syntax is created for model estimation. Optionally, syntax for constraints on the estimation model, in the `MODEL CONSTRAINTS` command, is created next. Ultimately, the parameter tables are combined to form character vectors containing the Mplus syntax to be exported (see "Details" of `powRICLPM` for more information on the `constraints` options).

**Naming conventions:** Details on the naming conventions can be found in the "Details" section of `powRICLPM`.

## Value

No return value, called for side effects.

## Examples

```
# Define population parameters for lagged effects
Phi <- matrix(c(.4, .1, .2, .3), ncol = 2, byrow = TRUE)

# Create and save Mplus model syntax
powRICLPM_Mplus(
  sample_size = c(400, 500),
  time_points = 3,
  ICC = 0.5,
  RI_cor = 0.3,
  Phi = Phi,
  within_cor = 0.3,
  reps = 10000,
  seed = 1234,
  save_path = tempdir()
)
```

---

# `print.powRICLPM` *Print* `powRICLPM` *object*

---

## Description

`print.powRICLPM` prints a table listing all experimental conditions contained in the `powRICLPM` object, as well as the frequency of the estimation problems that occurred in each.

## Usage

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

### Arguments

| | |
|---|---|
| x | A powRICLPM object. |
| ... | Argument not in use. |

### Value

No return value, called for side effects.

---

## summary.powRICLPM *Summarize setup and results from* powRICLPM *object*

---

### Description

S3 method for class powRICLPM. summary.powRICLPM summarizes and outputs the setup and results of the powRICLPM analysis. Depending on the arguments that are set, summary.powRICLPM provides a different summary (see "Details").

### Usage

```
## S3 method for class 'powRICLPM'
summary(
  object,
  ...,
  parameter = NULL,
  sample_size = NULL,
  time_points = NULL,
  ICC = NULL
)
```

### Arguments

| | |
|---|---|
| object | A powRICLPM object. |
| ... | (don't use) Additional arguments not affecting the summary produced. |
| parameter | Character string of length denoting the parameter to visualize the results for. |
| sample_size | (optional) An integer, denoting the sample size of the experimental condition of interest. |
| time_points | (optional) An integer, denoting the number of time points of the experimental condition of interest. |
| ICC | (optional) A double, denoting the proportion of variance at the between-unit level of the experimental condition of interest. |

## Details

summary.powRICLPM provides a different summary of the powRICLPM object, depending on the additional arguments that are set:

- When sample_size = ..., time_points = ..., and ICC = ... are set: Estimation information and results for all parameters of the experimental condition denoted by sample_size, time_points, and ICC.
- When parameter = "..." is set: Estimation information and results for a specific parameter across all experimental conditions.
- No additional arguments: Characteristics of the different experimental conditions are summarized, as well as session info (information that applies to each conditions, such the number of replications, etc.).

## Value

No return value, called for side effects.

## Examples

```
load(system.file("extdata", "power_preliminary.Rds", package = "powRICLPM"))

# Get setup of powRICLPM analysis and convergence issues
summary(power_preliminary)

# Performance measures for "wB2~wA1" parameter across experimental conditions
summary(power_preliminary, parameter = "wB2~wA1")

# Performance measures for all parameters, for specific experimental condition
summary(power_preliminary, sample_size = 600, time_points = 4, ICC = .5)
```

# Index