Package ‘SteppedPower’

July 7, 2021

Type Package

Title Power Calculation for Stepped Wedge Designs

Version 0.2.0

Description Tools for power and sample size
calculation as well as design diagnostics for
longitudinal mixed model settings, with a focus on stepped wedge designs.
All calculations are oracle estimates i.e. assume random effect variances
to be known (or guessed) in advance.
The method is introduced in Hussey and Hughes (2007) <doi:10.1016/j.cct.2006.05.007>,
extensions are discussed in Li et al. (2020) <doi:10.1177/0962280220932962>.

Imports Matrix, plotly, grDevices, stats, utils

Suggests knitr, rmarkdown, swCRTdesign, testthat, pwr

License MIT + file LICENSE

Encoding UTF-8

R topics documented:

alpha012_to_RandEff .............................................. 2
compute_wlsPower ............................................. 3
construct_CovBlk ............................................. 4
construct_CovMat ............................................ 5
construct_CovSubMat ....................................... 7
**Correlation structure: transform alpha to random effects**

**Description**

Correlation structure: transform alpha to random effects

**Usage**

```r
alpha012_to_RandEff(alpha012, sigResid = NULL, sigMarg = NULL)
```

**Arguments**

- `alpha012`: A vector or a list of length 3. Each list element must have the same dimension.
- `sigResid`: Residual standard deviation on individual level. Either residual sd or marginal sd needs to be specified.
- `sigMarg`: Marginal standard deviation on individual level. Either residual sd or marginal sd needs to be specified.

**Value**

A list containing four named elements (possibly matrices): random cluster intercept 'tau', random time effect 'gamma', random subject intercept and residual standard deviation.
compute_wlsPower

Examples

alpha012_to_RandEff(alpha012=c(.1,.1,.1), sigMarg=1)
alpha012_to_RandEff(alpha012=c(.1,.1,.1), sigResid=.9486833)

## The function is vectorised:
alpha012_to_RandEff(alpha012=list(matrix(c(0,.1,.1,.2), 2, 2),
                        matrix(c(0,0,.1,.2) , 2, 2),
                        matrix(c(0,0,.2,.2) , 2, 2)),
                        sigMarg=1)

compute_wlsPower  Compute power via weighted least squares

Description

This function is not intended to be used directly, but rather to be called by `wlsPower` - the main
function of this package. It expects the design matrix as an input argument `DesMat` and construct
the covariance matrix (if not given as well). These matrices are used to calculate the variance of the
treatment effect estimator which is then used to calculate the power to detect the assumed treatment
effect.

Usage

compute_wlsPower(
  DesMat,
  EffSize,
  sigma,
  tau = 0,
  eta = NULL,
  AR = NULL,
  rho = NULL,
  gamma = NULL,
  psi = NULL,
  N = NULL,
  CovMat = NULL,
  dfAdjust = "none",
  sig.level = 0.05,
  INDIV_LVL = FALSE,
  verbose = 1
)

Arguments

DesMat  object of class `DesMat`.
EffSize  raw effect, i.e. difference between mean under control and mean under intervention
construct_CovBlk

Description

Constructs the covariance matrix for multiple measurements of the same cluster. This function is usually called by `construct_CovMat` and is not designed to be used directly.

Usage

```r
construct_CovBlk(sigma, tau = NULL, eta = NULL, AR = NULL, rho = NULL)
```
construct_CovMat

Arguments

sigma   numeric (vector of length ‘timepoints’), residual error
tau     numeric (vector of length ‘timepoints’), standard deviation of random intercepts
eta     numeric (vector of length ‘timepoints’), standard deviation of random slope
AR      numeric, vector containing up to three values, each between 0 and 1. Defaults to NULL. It defines the AR(1)-correlation of random effects. The first element corresponds to the cluster intercept, the second to the treatment effect and the third to subject specific intercept. If only one element is provided, autocorrelation of all random effects is assumed to be the same. *Currently not compatible with ‘rho’!=0 !*
rho     numeric (scalar), correlation of ‘tau’ and ‘eta’

Value

a block of a covariance matrix, corresponding to intra-cluster covariance over time for one cluster

Examples

construct_CovBlk(sigma=rep(2,5), tau=rep(1,5))
construct_CovBlk(sigma=rep(2,5),
    tau=rep(.5,5), eta=c(0,0,1,1,1),
    AR=c(.5, 1))

construct_CovMat

Construct a Covariance Matrix

Description

constructs a (block diagonal) covariance matrix. This function calls ‘construct_CovBlk’ (or ‘construct_CovSubMat’ in case of repeated observations of the same individuals) for each block.

Usage

construct_CovMat(
    SumCl = NULL,
    timepoints = NULL,
    sigma,
    tau,
    eta = NULL,
    AR = NULL,
    rho = NULL,
    gamma = NULL,
    trtMat = NULL,
    N = NULL,
    CovBlk = NULL,
psi = NULL,
INDIV_LVL = FALSE
)

Arguments

SumCl total number of clusters
timepoints numeric (scalar or vector), number of timepoints (periods). If design is swd,
timepoints defaults to length(Cl)+1. Defaults to 1 for parallel designs.
sigma numeric, residual error of cluster means if no N given.
tau numeric, standard deviation of random intercepts
eta numeric (scalar or matrix), standard deviation of random slopes. If ‘eta’ is given
as scalar, ‘trtMat’ is needed as well.
AR numeric, vector containing up to three values, each between 0 and 1. Defaults
to NULL. It defines the AR(1)-correlation of random effects. The first element
corresponds to the cluster intercept, the second to the treatment effect and the
third to subject specific intercept. If only one element is provided, autocorrela-
tion of all random effects is assumed to be the same. *Currently not compatible
with ‘rho’!=0 !*
rho numeric (scalar), correlation of ‘tau’ and ‘eta’
gamma numeric (scalar), random time effect
trtMat a matrix of dimension *#Cluster* x *timepoints* as produced by the func-
tion ‘construct_trtMat’, indicating the cluster-periods that receive interventional
treatment. Defaults to NULL. If trtMat is given, the arguments ‘SumCl’ and
‘timepoints’ are ignored (!).
N numeric, number of individuals per cluster. Either a scalar, vector of length
#Clusters or a matrix of dimension #Clusters x timepoints. Defaults to 1 if not
passed.
CovBlk a matrix of dimension *timepoints* x *timepoints*.
psi numeric (scalar), random subject specific intercept. Leads to a closed cohort
setting
INDIV_LVL logical, should the computation be conducted on an individual level? This leads
to longer run time and is mainly for diagnostic purposes.

Value

a covariance matrix

Examples

## Two clusters, three timepoints,
## residual standard error sd=3, random slope sd=1.
construct_CovMat(SumCl=2, timepoints=3, sigma=3, tau=1)
##
## 
**construct_CovSubMat**

Construct a Block of the Covariance Matrix

### Description

Constructs the covariance matrix for multiple measurements of the same cluster if the same individuals are observed at all time periods. This function is not designed to be used directly.

### Usage

```r
construct_CovSubMat(
  N,
  timepoints,
  sigma,
  tau,
  eta = NULL,
  AR = NULL,
  rho = NULL,
  gamma = NULL,
  trtMat = NULL,
  psi = NULL,
  INDIV_LVL = FALSE
)
```

### Arguments

- **N**: Number of individuals per cluster
- **timepoints**: numeric (scalar or vector), number of timepoints (periods). If design is swd, timepoints defaults to length(Cl)+1. Defaults to 1 for parallel designs.
- **sigma**: numeric (vector of length ‘timepoints’), residual error
- **tau**: numeric (vector of length ‘timepoints’), standard deviation of random intercepts
- **eta**: numeric (vector of length ‘timepoints’), standard deviation of random slope
- **AR**: numeric, vector containing up to three values, each between 0 and 1. Defaults to NULL. It defines the AR(1)-correlation of random effects. The first element corresponds to the cluster intercept, the second to the treatment effect and the third to subject specific intercept. If only one element is provided, autocorrelation of all random effects is assumed to be the same. *Currently not compatible with ‘rho’!=0!*
construct_DesMat

Description

Constructs the design matrix with one column for every (fixed) parameter to be estimated and one row for every cluster for every timepoint. This function calls `construct_trtMat` to construct a matrix that indicates treatment status for each cluster at each timepoint. This is then transformed into the first column of the design matrix. `construct_CovMat` further calls `construct_timeAdjust` to get the fixed effect(s) of the timepoints.

Note: Unlike the usual notation, the treatment effect is in the first column (for easier access by higher level functions).

Usage

```r
construct_DesMat(
  Cl = NULL,
  trtDelay = NULL,
  dsntype = "SWD",
  timepoints = NULL,
  timeAdjust = "factor",
  period = NULL,
  trtmatrix = NULL,
  timeBlk = NULL,
  N = NULL,
  INDIV_LVL = FALSE
)
```
**construct_timeAdjust**

Construct the time period adjustment in the design matrix

### Description

Offers several options to adjust for secular trends.

### Arguments

**C1**
- integer (vector), number of clusters per sequence group (in SWD), or number in control and intervention (in parallel designs)

**trtDelay**
- numeric (possibly vector), value(s) between 0 and 1 specifying the proportion of intervention effect in the first, second, ... intervention phase.

**dsnType**
- character, defines the type of design. Options are "SWD", "parallel" and "parallel_baseline", defaults to "SWD".

**timepoints**
- numeric (scalar or vector), number of timepoints (periods). If design is swd, timepoints defaults to length(C1)+1. Defaults to 1 for parallel designs.

**timeAdjust**
- character, specifies adjustment for time periods. One of the following: "factor", "linear", "none", "periodic". Defaults to "factor".

**period**
- numeric (scalar)

**trtMatrix**
- an optional user defined matrix to define treatment allocation

**timeBlk**
- an optional user defined matrix that defines the time adjustment in one cluster. Is repeated for every cluster.

**N**
- numeric, number of individuals per cluster. Either a scalar, vector of length #Clusters or a matrix of dimension #Clusters x timepoints. Defaults to 1 if not passed.

**INDIV_LVL**
- logical, should the computation be conducted on an individual level? This leads to longer run time and is mainly for diagnostic purposes.

### Value

an object of class DesMat

### Examples

```r
construct_DesMat(Cl=c(2,0,1))
construct_DesMat(Cl=c(2,0,1), N=c(1,3,2))

## manually defined time adjustment (same as above)
timeBlock <- matrix(c(1,0,0,0,
                      1,1,0,0,
                      1,0,1,0,
                      1,0,0,1), 4, byrow=TRUE)
construct_DesMat(Cl=c(2,0,1), timeBlk=timeBlock)
```
construct_trtMat

Usage

construct_timeAdjust(
  Cl, 
  timepoints, 
  timeAdjust = "factor", 
  period = NULL, 
  timeBlk = NULL
)

Arguments

Cl integer (vector), number of clusters per sequence group (in SWD), or number in control and intervention (in parallel designs)
timepoints numeric (scalar or vector), number of timepoints (periods). If design is swd, timepoints defaults to length(Cl)+1. Defaults to 1 for parallel designs.
timeAdjust character, specifies adjustment for time periods. One of the following: "factor", "linear", "none", "periodic". Defaults to "factor".
period numeric (scalar)
timeBlk an optional user defined matrix that defines the time adjustment in one cluster. Is repeated for every cluster.

Value

a matrix with one row for every cluster at every timepoint and number of columns depending of adjustment type.

construct_trtMat Construct Treatment Matrix

Description

Constructs a matrix of '#cluster' rows and '#timepoint' columns, indicating treatment status in each cluster at each timepoint.

Usage

construct_trtMat(Cl, trtDelay, dsntype, timepoints = NULL)

Arguments

Cl integer (vector), number of clusters per sequence group (in SWD), or number in control and intervention (in parallel designs)
trtDelay numeric (possibly vector), value(s) between 0 and 1 specifying the proportion of intervention effect in the first (second ...) intervention phase.
dsntype character, defines the type of design. Options are "SWD", "parallel" and "parallel_baseline", defaults to "SWD".
timepoints numeric (scalar or vector), number of timepoints (periods). If design is swd, timepoints defaults to length(Cl)+1. Defaults to 1 for parallel designs.

Value

a matrix trtMat, where rows and columns correspond to cluster and timepoints, respectively

Examples

construct_trtMat(Cl=c(1,2,1), trtDelay=c(.2,.8), dsntype="SWD")

Description

plot.DesMat

Usage

## S3 method for class 'DesMat'
plot(x, show_colorbar = FALSE, ...)

Arguments

x An object of class ‘DesMat’
show_colorbar logical, should the colorbar be shown?
... Arguments to be passed to methods

Value

a plotly html widget, displaying the treatment status

Examples

x <- construct_DesMat(C=c(2,2,0,2,2,2),.5)
plot.wlsPower

**plot an object of class ‘wlsPower’**

### Description

Up to three plots (selectable by `which`) that visualise: influence of each cluster for each timepoint, the treatment status for each cluster for each timepoint and the covariance matrix. By default, only the first plot is returned.

### Usage

```r
## S3 method for class 'wlsPower'
plot(x, which = 1, show_colorbars = NULL, ...)
```

### Arguments

- **x**: object of class `wlsPower`
- **which**: Specify a subset of the numbers 1:3 to select plots
- **show_colorbars**: logical, should the colorbars be shown?
- **...**: Arguments to be passed to methods

### Value

A list of plotly html widgets

---

**plot_CovMat**

### Description

Currently not exported.

### Usage

```r
plot_CovMat(CovMat, show_colorbar = TRUE)
```

### Arguments

- **CovMat**: A covariance matrix (possibly in sparse matrix notation)
- **show_colorbar**: logical, should the colorbar be shown?

### Value

A plotly object
print.DesMat

Description
print.DesMat

Usage
## S3 method for class 'DesMat'
print(x, ...)

Arguments
x An object of class ‘DesMat
... Arguments to be passed to methods

Value
Messages with information about the design.

print.wlsPower

Description
Print an object of class ‘wlsPower’

Usage
## S3 method for class 'wlsPower'
print(x, ...)

Arguments
x object of class wlsPower
... Arguments to be passed to methods

Value
Messages, containing information about (at least) power and significance level
RandEff_to_alpha012  

**Description**

Correlation structure: transform random effects to alpha

**Usage**

RandEff_to_alpha012(sigResid, tau, gamma, psi)

**Arguments**

- **sigResid**: Residual standard deviation on individual level
- **tau**: Standard deviation of random cluster intercept
- **gamma**: Standard deviation of random time effect
- **psi**: Standard deviation of random subject specific intercept

**Value**

A list containing four named elements (possibly matrices): ‘alpha0’, ‘alpha1’, ‘alpha2’ specify a correlation structure and SigMarg denotes the marginal standard deviation

**Examples**

RandEff_to_alpha012(sigResid=sqrt(11), tau=4, gamma=3, psi=2)

## The function is vectorised:

RandEff_to_alpha012(sigResid = matrix(c(0,1,2,3,4,5), 2, 3),
                     tau = matrix(c(1,1,0,0,0), 2, 3),
                     gamma = matrix(c(0,0,1,0,0,1), 2, 3),
                     psi = matrix(c(0,1,0,1), 2, 3))

---

**Description**

SteppedPower offers tools for power and sample size calculation as well as design diagnostics for longitudinal mixed model settings, with a focus on stepped wedge designs. All calculations are oracle estimates i.e. assume random effect variances to be known (or guessed) in advance.

**Author(s)**

Philipp Mildenberger <pmildenb@uni-mainz.de>
tTestPwr

**Compute Power of a Wald Test**

**Description**

Computes the power of a scaled Wald test given a standard error, an effect size, the degrees of freedom of the t-distribution and a significance level. Computes the exact power, see second example.

**Usage**

```r
tTestPwr(d, se, df, sig.level = 0.05)
```

**Arguments**

- `d`: numeric, raw effect
- `se`: numeric, standard error
- `df`: numeric, degrees of freedom of the t-distribution
- `sig.level`: numeric, significance level, defaults to 0.05

**Value**

a scalar

**Examples**

```r
tTestPwr(4,1,10) ; tTestPwr(4,1,30) ; tTestPwr(4,1,Inf)
```

---

**VarClosed_Kasza**

*Closed formula for treatment variance in open cohort settings*

**Description**

From Kasza et al "Sample size and power calculations for open cohort longitudinal cluster randomized trials" 2020

**Usage**

```r
VarClosed_Kasza(trtMat, tau, gamma = 0, psi = 0, sigma, N, chi)
```
Arguments

- **trtMat**: a matrix `trtMat` to define treatment allocation, where rows and columns correspond to cluster and timepoints, respectively.
- **tau**: numeric, standard deviation of random intercepts.
- **gamma**: numeric, random time effect.
- **psi**: numeric, random subject specific intercept.
- **sigma**: numeric, residual error on subject level.
- **N**: numeric, number of individuals per cluster.
- **chi**: Attrition factor.

Value

numeric, variance of the estimator for treatment effect.

Examples

```r
# test setting, from Hussey&Hughes 2007
trtMat <- construct_DesMat(c(6,6,6,6))$trtMat
tau <- .025; sigma <- sqrt(.041*.959); N <- 100;
gamma <- 0.01; psi <- .1; chi <- .7

tmp <- VarClosed_Kasza(trtMat, tau=tau, sigma=sigma, gamma=0, psi=0, N=N, chi=0)
tTestPwr((.05-.032), sqrt(tmp), df = Inf)
wlsPower(CL = rep(6,4), N=N, mu0=.05, mu1=.032, verbose=0,
          sigma=sigma, gamma=0, tau=tau, psi=0)

tmp <- VarClosed_Kasza(trtMat, tau=tau, sigma=sigma, gamma=gamma, psi=psi, N=N, chi=0)
tTestPwr((.05-.032), sqrt(tmp), df = Inf)
wlsPower(CL = rep(6,4), N=N, mu0=.05, mu1=.032, verbose=0,
          sigma=sigma, gamma=gamma, tau=tau, psi=psi)

tmp <- VarClosed_Kasza(trtMat, tau=tau, sigma=sigma, gamma=gamma, psi=psi, N=N, chi=1)
tTestPwr((.05-.032), sqrt(tmp), df = Inf)
wlsPower(CL = rep(6,4), N=N, mu0=.05, mu1=.032, verbose=0,
          sigma=sigma, gamma=sqrt(gamma^2+psi^2/N), tau=tau, psi=0)

tmp <- VarClosed_Kasza(trtMat, tau=tau, sigma=sigma, gamma=gamma, psi=psi, N=N, chi=chi)
tTestPwr((.05-.032), sqrt(tmp), df = Inf)
wlsPower(CL = rep(6,4), N=N, mu0=.05, mu1=.032, verbose=0,
          sigma=sigma, gamma=sqrt(gamma^2+chi*psi^2/N), tau=tau, psi=sqrt(1-chi)*psi)
```

---

**VarClosed_Li**

*Closed formula for treatment variance, with proportional decay*

Description

From Li et al "Design and analysis considerations for cohort stepped wedge cluster randomized trials with a decay correlation structure***
Usage

VarClosed_Li(trtMat, tau, psi, N, AR)

Arguments

- **trtMat**: a matrix to define treatment allocation, where rows and columns correspond to cluster and timepoints, respectively.
- **tau**: numeric, standard deviation of random intercepts.
- **psi**: numeric, random subject-specific intercept.
- **N**: numeric, number of individuals per cluster.
- **AR**: numeric (scalar), defines the AR(1)-correlation of random effects.

Value

numeric, variance of the estimator for treatment effect

Examples

```r
## test setting, from Hussey&Hughes 2007 ####
trtMat <- construct_DesMat(c(6,6,6,6))$trtMat
tau <- .025 ; N <- 100 ; psi <- .1 ; AR <- .6
tmp <- VarClosed_Li(trtMat, tau=tau, psi=psi, N=N, AR=AR)
tTestPwr((.05-.032), se=sqrt(tmp), Inf)
wlsPower(Cl=rep(6,4), mu0=.05, mu1=.032, AR=AR,
          tau=tau, N=N, sigma=0, psi=psi, verbose=0)
```

wlsPower

*Compute power via weighted least squares*

Description

This is the main function of the SteppedPower package. It calls the constructor functions for the design matrix and covariance matrix, and then calculates the variance of the intervention effect estimator. The latter is then used to compute the power of a Wald test of a (given) intervention effect.

Usage

```r
wlsPower(
  Cl = NULL,
  timepoints = NULL,
  DesMat = NULL,
  trtDelay = NULL,
  incomplete = NULL,
  timeAdjust = "factor",
  period = NULL,
)```
dsntype = "SWD",
mu0,
mu1,
marginal_mu = FALSE,
sigma = NULL,
tau = NULL,
etra = NULL,
AR = NULL,
rho = NULL,
gamma = NULL,
psi = NULL,
alpha_0_1_2 = NULL,
CovMat = NULL,
N = NULL,
power = NULL,
family = "gaussian",
N_range = c(1, 1000),
sig.level = 0.05,
dfAdjust = "none",
INDIV_LVL = FALSE,
verbose = 1
)

Arguments

Cl
integer (vector), number of clusters per sequence group (in SWD), or number in control and intervention (in parallel designs)

timepoints
numeric (scalar or vector), number of timepoints (periods). If design is swd, timepoints defaults to length(Cl)+1. Defaults to 1 for parallel designs.

DesMat
Either an object of class `DesMat` or a matrix indicating the treatment status for each cluster at each timepoint. If supplied, 'timepoints','Cl','trtDelay' are ignored.

trtDelay
numeric (possibly vector), value(s) between 0 and 1 specifying the proportion of intervention effect in the first (second ... ) intervention phase.

incomplete
integer, either a scalar (only for SWD) or a matrix. A vector defines the number of periods before and after the switch from control to intervention that are observed. A matrix consists of 1’s for observed clusterperiods and 0’s for unobserved clusterperiods.

timeAdjust
character, specifies adjustment for time periods. One of the following: "factor", "linear", "none", "periodic". Defaults to "factor".

period
numeric (scalar)

dsntype
character, defines the type of design. Options are "SWD", "parallel" and "parallel_baseline", defaults to "SWD".

mu0
numeric (scalar), defaults to "SWD".

mu1
numeric (scalar), mean under treatment
marginal_mu  logical. Only relevant for non-gaussian outcome. Indicates whether \( \mu_0 \) and \( \mu_1 \) are to be interpreted as marginal prevalence under control and under treatment, respectively, or whether they denote the prevalence conditional on random effects being 0 (it defaults to the latter). *(experimental!)*

sigma  numeric, residual error of cluster means if no N given.

tau  numeric, standard deviation of random intercepts

eta  numeric (scalar or matrix), standard deviation of random slopes. If ‘eta’ is given as scalar, ‘trtMat’ is needed as well.

AR  numeric, vector containing up to three values, each between 0 and 1. Defaults to NULL. It defines the AR(1)-correlation of random effects. The first element corresponds to the cluster intercept, the second to the treatment effect and the third to subject specific intercept. If only one element is provided, autocorrelation of all random effects is assumed to be the same. *Currently not compatible with ‘rho’!=0 !*

rho  numeric (scalar), correlation of ‘tau’ and ‘eta’

gamma  numeric (scalar), random time effect

psi  numeric (scalar), random subject specific intercept. Leads to a closed cohort setting

alpha_0_1_2  numeric vector or list of length 2 or 3, that consists of alpha_0, alpha_1 and alpha_2. Can be used instead of random effects to define the correlation structure, following Li et al. (2018). When omitting alpha_2, this describes a cross-sectional design, where alpha_0 and alpha_1 define the intracluster correlation and cluster autocorrelation, respectively - as defined by Hooper et al. (2016).

CovMat  numeric, a positive-semidefinite matrix with (#Clusters · timepoints) rows and columns. If ‘CovMat’ is given, ‘sigma’, ‘tau’, ‘eta’, ‘rho’, ‘gamma’ and ‘psi’ as well as ‘alpha_0_1_2’ must be NULL.

N  numeric, number of individuals per cluster. Either a scalar, vector of length #Clusters or a matrix of dimension #Clusters x timepoints. Defaults to 1 if not passed.

power  numeric, a specified target power. If supplied, the minimal ‘N’ is returned.

family  character, distribution family. One of ”gaussian”, ”binomial”. Defaults to ”gaussian”

N_range  numeric, vector specifying the lower and upper bound for ‘N’, ignored if ‘power’ is NULL.

sig.level  numeric (scalar), significance level, defaults to 0.05

dfAdjust  character, one of the following: ”none”, ”between-within”, ”containment”, ”residual”.

INDIV_LVL  logical, should the computation be conducted on an individual level? This leads to longer run time and is mainly for diagnostic purposes.

verbose  integer, how much information should the function return?

**Details**

Let \( \theta := \mu_1 - \mu_0 \) the treatment effect under investigation. The variance of the treatment effect estimator \( \hat{\theta} \) can then be estimated via weighted least squares (see also vignette ’Getting Started’).
Value

The return depends on the `verbose` parameter. If `verbose`=0, only the power is returned. If `verbose`=1 (the default), a list containing power and the parameters of the specific setting is returned. If requested (by `verbose`=2) this list also contains relevant matrices.

Examples

```r
## See also vignette for more examples
##
## ... stepped wedge design with 5 Clusters in 5 sequences,
## residual standard deviation 2,
## cluster effect sd = 0.33, and 10 individuals per cluster.
## Further, let the mean under the null alternative hypothesis 0 and 1,
## respectively.
wlsPower(mu0=0, mu1=1, Cl=rep(1,5), sigma=2, tau=0.33, N=10)
##
## ... with auto-regressive cluster effect `AR=0.7`.
wlsPower(mu0=0, mu1=1, Cl=rep(1,5), sigma=2, tau=0.33, AR=0.7, N=10)
##
## ... with varying cluster size
wlsPower(mu0=0, mu1=1, Cl=rep(1,5), sigma=2, tau=0.33, N=c(12,8,10,9,14))
wlsPower(mu0=0, mu1=1, Cl=rep(1,5), sigma=2, tau=0.33, N=matrix(c(12,8,10,9,14,
  11,8,10,9,13,
  11,7,11,8,12,
  10,7,10,8,11,
  9,7, 9,7,11,
  9,6, 8,7,11),5,6))
##
## ... with random treatment effect (with standard deviation 0.2),
## which is correlated with the cluster effect with `rho`=0.25.
wlsPower(mu0=0, mu1=1, Cl=rep(1,5), sigma=2, tau=0.33, eta=.2, rho=.25, N=10)
##
## ... with missing observations (a.k.a. incomplete stepped wedge design)
wlsPower(mu0=0, mu1=1, Cl=rep(1,5), sigma=2, tau=0.33, N=10, incomplete=3)
wlsPower(mu0=0, mu1=1, Cl=rep(1,5), sigma=2, tau=0.33, N=10,
  incomplete=matrix(c(1,1,1,0,0,
    1,1,1,1,1,
    1,1,1,1,1,
    0,1,1,1,1,
    0,0,1,1,1),5,6))
## -> the same.
##
## ... with two levels of clustering. This arises if the patients are
## observed over the whole study period
## (often referred to as closed cohort design) or if subclusters exist
```
(such as wards within clinics). For

```r
mod_aggr <- wlsPower(mu0=0, mu1=1, Cl=rep(1,5),
  sigma=2, tau=0.33, psi=.25,
  N=10, incomplete=3, verbose=2)

mod_indiv <- wlsPower(mu0=0, mu1=1, Cl=rep(1,5),
  sigma=2, tau=0.33, psi=.25,
  N=10, incomplete=3, verbose=2, INDIV_LVL=TRUE)
```

```r
mod_aggr
mod_indiv
```

## Compare covariance matrices of first cluster

```r
```

## stepped wedge design with 5 Clusters in 5 sequences, residual sd = 2,
## cluster effect sd = 0.33. How many Individuals are needed to achieve a
## power of 80% ?

```r
wlsPower(mu0=0, mu1=1, Cl=rep(1,5), sigma=2, tau=0.33, power=.8)
```

## ... How many are needed if we have a closed cohort design with a random
## individuum effect of .7?

```r
wlsPower(mu0=0, mu1=1, Cl=rep(1,5), sigma=2, tau=0.33, psi=.7, power=.8)
```

## longitudinal parallel design, with 5 time periods, 3 clusters in treatment
## and control arm each.

```r
wlsPower(mu0=0, mu1=1, Cl=c(3,3), sigma=2, tau=0.33, N=10,
  dsntype="parallel", timepoints=5)
```

## ... with one baseline period and four parallel periods

```r
wlsPower(mu0=0, mu1=1, Cl=c(3,3), sigma=2, tau=0.33, N=10,
  dsntype="parallel_baseline", timepoints=c(1,4))
```

## cross-over design with two timepoints before and two after the switch

```r
wlsPower(mu0=0, mu1=1, Cl=c(3,3), sigma=2, tau=0.33, N=10,
  dsntype="crossover", timepoints=c(2,2))
```

## stepped wedge design with 32 Individuals in 8 sequences, binomial outcome,
## 50% incidence under control, 25% incidence under interventional treatment.
## cluster effect sd = 0.5 (ICC of 1/3 under control),
## every individual is its own cluster.

```r
wlsPower(mu0=0.5, mu1=0.25, Cl=rep(4,8), tau=0.5, N=1,
  family="binomial")
```

## ... with incidences defined conditional on cluster effect=0

```r
wlsPower(mu0=0.5, mu1=0.25, Cl=rep(4,8), tau=0.5, N=1,
  family="binomial")
```

## ... with marginally defined proportions

```r
wlsPower(mu0=0.5, mu1=0.25, Cl=rep(4,8), tau=0.5, N=1,
  family="binomial")
```
wlsPower

family="binomial", marginal_mu=TRUE)
Index

alpha012_to_RandEff, 2
compute_wlsPower, 3
construct_CovBlk, 4
construct_CovMat, 5
construct_CovSubMat, 7
construct_DesMat, 8
construct_timeAdjust, 9
construct_trtMat, 10
plot.DesMat, 11
plot.wlsPower, 12
plot_CovMat, 12
print.DesMat, 13
print.wlsPower, 13
RandEff_to_alpha012, 14
SteppedPower-pkg, 14
tTestPwr, 15
VarClosed_Kasza, 15
VarClosed_Li, 16
wlsPower, 17