Package 'swCRTdesign'

October 29, 2026

Type Package

Title Stepped Wedge Cluster Randomized Trial (SW CRT) Design

Version 4.2

Description A set of tools for examining the design and analysis aspects of stepped wedge cluster randomized trials (SW CRT) based on a repeated cross-sectional or cohort sampling scheme (Hussey MA and Hughes JP (2007) Contemporary Clinical Trials 28:182-191).

License GPL-2

Encoding UTF-8

Imports stats, lme4, lmerTest

NeedsCompilation no

Author Jim Hughes [aut, cre],
Navneet R. Hakhu [aut],
Emily Voldal [aut],
Fan Xia [aut]

Maintainer Jim Hughes <jphughes@uw.edu>

Repository CRAN

Date/Publication 2026-10-29 00:10:02 UTC

Contents

swCRTdesign-package 2
blkDiag 3
Release Notes 4
swDsn 4
swGlmPwr 8
swGlmSiz 14
swPlot 15
swPwr 17
swSim 24
swSim2 27
swSimPwr 31

swSiz 34
swSummary 36

Index **39**

`swCRTdesign-package` *Stepped Wedge Cluster Randomized Trial (SW CRT) Design*

Description

This package includes functions for the design and analysis of stepped wedge cluster randomized trials. For additional guidance, see (Voldal EC, Hakhu NR, Xia F, Heagerty PJ, Hughes JP. swCRTdesign: An R package for stepped wedge trial design and analysis. Computer Methods and Programs in Biomedicine 2020;196:105514. Nine primary functions - `swPwr`, `swSiz`, `swGlmPwr`, `swGlmSiz`, `swSimPwr`, `swSim`, `swSim2`, `swSummary`, and `swPlot` - and two support functions - `blkDiag` and `swDsn` - are included.

The `blkDiag` function creates a block diagonal matrix from a specified array or list of block-matrices. The `swDsn` function creates a stepped wedge (SW) design object based on specified information on clusters, time points, and the arms of the cluster randomized trial (CRT).

The `swPwr` function computes the (two-sided) power of treatment effect (θ) for the specified SW CRT design via weighted least squares (WLS), where the response/outcome of interest is assumed to come from a mixed effects model with normal or binomial errors, linear link and random time effects and (possibly correlated) random intercepts and random treatment effects. The random time effects apply to all time points, and time is treated as categorical. An exponentially decaying autocorrelation structure is also supported. `swSiz` is a wrapper function for `swPwr` and allows one to compute the cluster-period size for a given design and power.

`swGlmPwr` has functionality similar to `swPwr` and does power calculations using the generalized linear mixed model framework (Xia et al, 2019); `swGlmSiz` is the corresponding wrapper function for computing cluster-period size.

`swSimPwr` simulates data and runs analyses using the linear mixed model or generalized linear mixed model framework to compute power via simulation. `swPwr`, `swSiz`, `swGlmPwr`, `swGlmSiz` and `swSimPwr` provide power/sample size calculations for both an immediate treatment (IT) model and an exposure time indicator (ETI) model (Kenny et al, 2022) and can handle cross-sectional or closed cohort designs.

The `swSim` function generates individual-level data consisting of response, treatment, time, time on treatment and cluster id variables (and individual id for a closed cohort) based on a specified SW CRT design. `swSim2` extends the functionality of `swSim` by allowing one to simulate data with an exponentially decaying autocorrelation structure as described in Kasza et al. (2019). In addition, `swSim2` uses the glmmTMB function simulate_new to simulate datasets. `swSim` has been retained for backwards compatability.

The `swSummary` function computes the mean, sum, or number of non-missing response values for clusters separately or aggregated by wave at each time point from stepped wedge data that includes, at least, response, treatment, time, and cluster variables.

The `swPlot` function plots mean response as a combined or separate plot, for waves and clusters.

Some features of the package are also available as a shiny app, available online (https://swcrtdesign.shinyapps.io/stepped_wed
or to download and run locally (https://github.com/swCRTdesign/Stepped-wedge-power-calculation).

Details

Package:	swCRTdesign
Type:	Package
Version:	4.1
Date:	2025-08-18
License:	GPL (>= 2)

Author(s)

James P Hughes, Navneet R Hakhu, Emily C Voldal, and Fan Xia

Maintainer: James P Hughes <jphughes@uw.edu>

References

Hussey MA, Hughes JP. Design and analysis of stepped wedge cluster randomized trials. Contemporary Clinical Trials 2007;28:182-191.

Kenny A, Voldal E, Xia F, Heagerty PJ, Hughes JP. Analysis of stepped wedge cluster randomized trials in the presence of a time-varying treatment effect. Statistics in Medicine, in press, 2022.

Voldal EC, Hakhu NR, Xia F, Heagerty PJ, Hughes JP. swCRTdesign: An R package for stepped wedge trial design and analysis. Computer Methods and Programs in Biomedicine 2020;196:105514.

Xia F, Hughes JP, Voldal EC, Heagerty PJ. Power and sample size calculation for stepped-wedge designs with discrete outcomes. Trials. 2021 Dec;22(1):598.

---

blkDiag *Block Diagonal Matrix Generator*

Description

`blkDiag` returns block diagonal matrix based on specified square blocks (either as an array or a list).

Usage

```
blkDiag(z)
```

Arguments

`z`	numeric (array or list): User-specified matrices to be combined into one block diagonal matrix.

Value

numeric (matrix): `blkDiag` gives a block diagonal matrix.

Author(s)

James P Hughes and Navneet R Hakhu

Examples

```
library(swCRTdesign)
# Example 1 (input: array)
blkDiag.Ex1.array <- blkDiag( z=array(1:12, c(2,2,3)) )
blkDiag.Ex1.array

# Example 2 (input: list)
blkDiag.Ex2.list <- blkDiag( z=list(matrix(1,2,2), diag(2,3), matrix(3,4,4)) )
blkDiag.Ex2.list
```

`Release Notes` *Release notes for v4.2*

Description

This new version has been released because I received an email indicating that glmmTMB might be removed from CRAN.

This affects the functions swSimPwr (which calls glmmTMB) and swSim2 (which calls simulate_new in the glmmTMB package) in the swCRTdesign package. glmmTMB was used because of bugs in the earlier lme4 implementation of the ar() function. These bugs have been fixed and I have now updated the swSimPwr function to call lmer/glmer instead of glMMTMB and swSim2 to call simulate instead of simulate_new.

If you are using autoregressive models (for power or simulation) it is important that you have the latest version of the lme4 library, which can be downloaded from GitHub:

remotes::install_github("lme4/lme4")

`swDsn` *Study design of Stepped Wedge Cluster Randomized Trial (SW CRT)*

Description

`swDsn` returns a SW CRT study design object.

All calls to `swDsn` require a `clusters` argument, a vector that specifies the number of clusters in each wave (all clusters that start the intervention at a given time point are collectively referred to as a wave or sequence). Used by itself, `clusters` specifies a classic stepped wedge design with number of waves equal to the length of `clusters` and number of time points equal to number of waves plus one. Further design modifications may be specified using the `extra.ctl.time`, `extra.trt.time`, and `all.ctl.time0` arguments. Fractional treatment effects may be specified for each time after the intervention is introduced using the `tx.effect.frac` argument.

Alternatively, the swBlk argument can be used to specify a completely arbitrary stepped wedge design with multiple treatment levels and/or periods with no data collection (alternatively, periods with no data collection can be defined in the sample size arguments of swPwr, swGlmPwr and swSimPwr).

One can choose to use the extra.ctl.time, extra.trt.time, and all.ctl.time0 arguments or the swBlk argument, but not both. Also, tx.effect.frac may not be used with the swBlk argument.

swDsn is used by other functions in this package.

Usage

```
swDsn(clusters, tx.effect.frac=1, extra.ctl.time=0, extra.trt.time=0,
all.ctl.time0=TRUE, swBlk, extra.time=NULL)
```

Arguments

clusters	integer (vector): Number of clusters for each wave (e.g. c(6,6,6,6) specifies four waves with 6 clusters in each wave). A value of 0 in the vector means that no clusters introduce the intervention at a given time (see examples). Note that 0 clusters in a wave is not allowed when using the swBlk specification.
tx.effect.frac	numeric (scalar or vector): Relative treatment effect upon crossing over from control. Note that this is not the treatment effect (the treatment effect is specified in the call to other functions such as swPwr or swSim). If tx.effect.frac is set to a scalar with value of 1, the standard SW CRT treatment effect will be presumed. If tx.effect.frac is set to a scalar with a fractional value between 0 and 1, then only the first time point upon crossing over from control will have fractional relative treatment effect; the remaining exposure times will have a relative treatment effect of 1. If a vector of fractional treatment effect is specified, each element of the vector corresponds to the (fractional) treatment effect upon crossing over from control; if the length of the vector is less than total number of exposure times, the remaining time points will have a relative treatment effect value of 1; if the length of the vector is greater than total number of time points after crossing over, not all elements of tx.effect.frac will be used. The default value is (scalar) 1.
extra.ctl.time	integer (scalar): Number of additional time steps during the control period beyond the standard SW CRT design. The default value is 0.
extra.trt.time	integer (scalar): Number of additional time steps during the treatment period beyond the standard SW CRT design. The default value is 0.
all.ctl.time0	logical: If TRUE, all clusters receive control at the first time point. If FALSE, clusters in the first wave (i.e., the first element of clusters) receive intervention at the first time. The default is TRUE.
swBlk	integer (matrix): matrix with number of rows equal to the number of waves and number of columns equal to the total number of time periods of the study. Each row of swBlk corresponds to a wave and describes the introduction and levels of the intervention. Each element of swBlk is either NA, meaning no data will be collected in that cluster-time period, or an integer. Integers correspond to levels of the intervention where 0 is the control condition and values >0 correspond to various levels of the intervention. A classic SW design includes only 2 values

(0 and 1). More generally, one can specify integers corresponding to different levels of the intervention (e.g., a learning period, full intervention effect period and followup period could be coded as 1, 2 and 3, respectively). Alternatively, level 2 might denote the introduction of a second intervention that replaces the first, or is added to the first. A separate parameter is assigned for each intervention level so power can be computed for comparisons between any level and the control, or between different levels.

Note that the ETI model of Kenny et al (2022) implies a different intervention level for each exposure lag time. However, if you plan to use an ETI-based estimator of the treatment effect, then you should specify a standard 2-value design (0/1) in `swDsn` and use the H argument in `swPwr` or `swGlmPwr` to determine power for the estimator of interest. Also, note that NA periods that are interspersed with intervention periods (e.g. 0, 1, NA, 1) may cause unpredictable results for power calculations for ETI-based estimators since the lag for periods after the NA is ill-defined; NA periods that occur after the last intervention lag period of interest (e.g. 0, 1, 1, NA) are okay, however.

`extra.time` integer (scalar): deprecated; if used, replaces extra.trt.time.

Details

The `clusters` argument is required. After that, a design may be specified using the `tx.effect.frac`, `extra.ctrl.time`, `extra.trt.time`, `all.ctl.time0` arguments or the `swBlk` argument:

```
swDsn(clusters, tx.effect.frac=1, extra.ctl.time=0, extra.trt.time=0, all.ctl.time0=TRUE)
```

or

```
swDsn(clusters, swBlk)
```

Value

numeric (list): Returns the following user-specified and computed objects

`swDsn` numeric (matrix): schematic representation of the specified SW CRT design. Number of clusters is equal to the number of rows of the matrix and total number of time intervals is equal to the number of columns of the matrix. `swDsn[i,j]` gives the intervention status for cluster `i` at time `j` and has possible values NA (no data), 0 (control), or >=1 (intervention levels).

`swDsn.unique.clusters` numeric (matrix): Truncated SW CRT design of interest, with one row for each wave.

`n.waves` numeric (scalar): Number of waves for the SW CRT design of interest.

`clusters` numeric (vector): Number of clusters for each wave for the SW CRT design of interest.

`n.clusters` numeric (scalar): Total number of clusters for the SW CRT design of interest.

`tx.effect.frac` numeric (scalar or vector): Fractional treatment effect for time points upon crossing over from control of SW CRT design of interest.

`total.time` numeric (scalar): Total number of time points for the SW CRT design of interest.

`nTxLev` numeric (integer): Number of treatment levels.

`TxLev` numeric (integer): Vector of unique treatment levels.

Author(s)

James P Hughes and Navneet R Hakhu

References

Hussey MA, Hughes JP. Design and analysis of stepped wedge cluster randomized trials. Contemporary Clinical Trials 2007;28:182-191.

Voldal EC, Hakhu NR, Xia, F, Heagerty PJ, Hughes JP. swCRTdesign: An R Package for Stepped Wedge Trial Design and Analysis. Computer Methods and Programs in Biomedicine 2020;196:105514.

Examples

```
# Example 1 (Equal clusters per wave, standard SW design)
swDsn.Ex1.std <- swDsn( clusters=c(3,3,3) )
swDsn.Ex1.std$swDsn

# Example 2 (Equal clusters per wave, extended SW design)
swDsn.Ex1.extend <- swDsn( clusters=c(3,3,3), extra.trt.time=2 )
swDsn.Ex1.extend$swDsn

# Example 3 (Equal clusters per wave, not all ctl at time 0, "standard" for time SW design)
swDsn.Ex1.std.noAllctl <- swDsn( clusters=c(3,3,3), all.ctl.time0=FALSE )
swDsn.Ex1.std.noAllctl$swDsn

# Example 4 (Equal clusters per wave, not all ctl at time 0, extended SW design)
swDsn.Ex1.extend.noAllctl <- swDsn( clusters=c(3,3,3), extra.trt.time=2, all.ctl.time0=FALSE )
swDsn.Ex1.extend.noAllctl$swDsn

# Example 5 (Unequal clusters per wave, standard SW design)
swDsn.Ex1.std.unequal <- swDsn( clusters=c(3,0,2) )
swDsn.Ex1.std.unequal$swDsn

# Example 6 (Unequal clusters per wave, extended SW design)
swDsn.Ex1.extend.unequal <- swDsn( clusters=c(3,0,2), extra.trt.time=2 )
swDsn.Ex1.extend.unequal$swDsn

# Example 7 (Unequal clusters per wave, extended SW design)
swDsn.Ex1.extend.unequal.varyTxEffect <- swDsn( clusters=c(3,0,2), tx.effect.frac=c(0.8,0.9,1.0),
extra.trt.time=2 )
swDsn.Ex1.extend.unequal.varyTxEffect$swDsn

#Example 8 (Equal clusters per wave, extra control time and extra treatment time)
swDsn.Ex1.extendctrl.extendtrt <- swDsn(c(5,5,5,5,5),extra.ctl.time=3,extra.trt.time=5)
swDsn.Ex1.extendctrl.extendtrt$swDsn

# Examples with swBlk
# Example 9 (Equal clusters per wave, standard SW design)
mat9 = matrix(c(0,1,1,1,1,0,0,1,1,1,0,0,0,1,1,0,0,0,0,1),4,5,byrow=TRUE)
swDsn.Ex9 <- swDsn(c(6,6,6,6),swBlk=mat9)
swDsn.Ex9$swDsn
```

```
# Example 10 (Unequal clusters per wave, periods with no data collection)
mat10 = matrix(c(0,1,1,1,1,NA,0,1,1,1,NA,NA,0,1,1,NA,NA,NA,0,1),4,5,byrow=TRUE)
swDsn.Ex10 <- swDsn(c(5,6,6,5),swBlk=mat10)
swDsn.Ex10$swDsn

# Example 11 (Unequal clusters per wave, periods with no data collection,
# multiple intervention levels)
mat11 = matrix(c(0,1,2,2,2,2,NA,0,1,2,2,2,NA,NA,0,1,2,2,NA,NA,NA,0,1,2),4,6,byrow=TRUE)
swDsn.Ex11 <- swDsn(c(5,6,6,5),swBlk=mat11)
swDsn.Ex11$swDsn

# Example 12 (Dogleg design)
swDsn.Ex12 <- swDsn(c(4,4,4), swBlk=matrix(c(1,NA,0,1,NA,0),3,2,byrow=TRUE))
swDsn.Ex12$swDsn
```

`swGlmPwr` *Power of Stepped Wedge Cluster Randomized Trial with Discrete Outcomes*

Description

`swGlmPwr` returns (two-sided) power of the treatment effect for the specified SW CRT design in the context of generalized linear models by adopting the Laplace approximation detailed in Breslow and Clayton (1993) to obtain the covariance matrix of the estimated parameters. The response/outcome of interest can be binomial (logit link only) or Poisson (log link only) distributed. A cross-sectional or closed cohort sampling scheme can be specified. The outcome is assumed to come from a model with fixed treatment effect(s) (using an immediate treatment (IT) or exposure time indicator (ETI) model - see Kenny et al (2022)), fixed time effect, random intercepts, random treatment effects, random cluster-specific time effects and, in the case of closed cohort sampling, an individual random effect. The coefficients for fixed effects can be specified using `fixed.intercept`, `fixed.treatment.effect`, and `fixed.time.effect`. Variance components can be specified using `tau`, `eta`, `rho`, `gamma` and `zeta`.

Usage

```
swGlmPwr(design, distn, n, fixed.intercept, fixed.treatment.effect,
fixed.time.effect, H = NULL, tau = 0, eta = 0, rho = 0, gamma = 0, zeta = 0, ar=1,
alpha=0.05, retDATA = FALSE, silent=FALSE)
```

Arguments

`design`	list: A stepped wedge design object, typically from `swDsn`, that includes at least the following components: swDsn, swDsn.unique.clusters, clusters, n.clusters, total.time, nTxLev, TxLev. Fractional treatment effects specified in `swDsn` are used in the calculation.
`distn`	character: Distribution assumed (binomial or Poisson). "binomial" implies binomial outcomes with logit link and "poisson" implies Poisson outcome with log link. ***NOTE: It is the users responsibility to make sure specified parameters

(fixed.intercept, fixed.treatment effect, fixed.time effect, tau, eta, rho, gamma, zeta) are ALL on SAME scale as specified link function; see example.***

`n` integer (scalar, vector, or matrix): Number of observations: (scalar) for all clusters and all time points that are not NA in design$swDsn; (vector) for each cluster at all time points that are not NA in design$swDsn; and (matrix) for each cluster at each time point, where rows correspond to clusters and columns correspond to time. `n` can also be used to specify a design with transition periods (a transition period is a period, typically, immediately after treatment is introduced were no data are collected). Simply define `n` as a matrix with a sample size of 0 during every transition period. This is equivalent to putting an NA in the corresponding cluster=period in design$swDsn.

`fixed.intercept`
numeric (scalar): Intercept for the fixed effect on canonical scales (logit for binomial outcomes and log for Poisson outcomes). It is the mean outcome under the control condition in the first time point transformed to the canonical scales.

`fixed.treatment.effect`
numeric (scalar, vector): Gives the coefficients for the treatment effect on the canonical scale (logit for binomial outcomes and log for Poisson outcomes).

If `H` = NULL then an IT model is assumed and `fixed.treatment.effect` gives the coefficient(s) for the `design$nTxLev` treatment effect(s). If `tx.effect.frac` is specified in `design` then the assumed treatment effect at exposure time k is `tx.effect.frac[k]*fixed.treatment.effect`.

If `H` is non-NULL then an ETI model is assumed and `fixed.treatment.effect` is a vector as long as the longest treatment effect exposure time; in a classic stepped wedge design, the longest exposure time is the number of time periods minus one. Note that the length of `fixed.treatment.effect` must agree with the longest exposure time by design, regardless of the the sample size (`n`)

`fixed.time.effect`
numeric(scalar, vector): Coefficients for the time (as dummy variables) in the fixed effect model on canonical scales (logit for binomial outcomes and log for Poisson outcomes). The first time point is always used as reference. Specify a common time effect for all time points after the first (scalar) or differnt time effects for each time point after the first (vector of length (total time-1)).

`H` numeric (vector): If NULL, then swGlmPwr assumes an immediate, constant treatment effect (IT) model. If not NULL, then an exposure time indicator (ETI) model is assumed and H should be a vector as long as the longest exposure time (in a classic SW design, the longest exposure period is the number of time periods minus one). H specifies the desired linear combination of fixed.treatment.effect. For example, in a stepped wedge trial with 5 time periods and four exposure times, H = rep(.25,4) gives the average treatment effect over the four exposure times; H = c(0,0,.5,.5) ignores the first two periods after the intervention is introduced and averages the remaining periods. Typically, the sum of H is 1.0; if not, it is renormalized to sum to 1.0. H can only be specified when there is a single intervention type (i.e. design$swDsn includes only NA,0,1; see cautions in help for `swDsn` about including NA periods when H is used). Do not use an ETI model (non-null `H`) if you specified a fractional treatment effect in `swDsn`. See the details section for additional notes regarding incomplete designs.

tau	numeric (scalar): Standard deviation of random intercepts on canonical scales (logit for binomial outcomes and log for Poisson outcomes).
eta	numeric (scalar): Standard deviation of random treatment effects on canonical scales (logit for binomial outcomes and log for Poisson outcomes).
rho	numeric (scalar): Correlation between random intercepts and random treatment effects on canonical scales (logit for binomial outcomes and log for Poisson outcomes).
gamma	numeric (scalar): Standard deviation of random time effects on canonical scales (logit for binomial outcomes and log for Poisson outcomes).
zeta	numeric (scalar): Standard deviation of individual effects for closed cohort sampling on canonical scales (logit for binomial outcomes and log for Poisson outcomes). Default is 0 which implies cross-sectional sampling. Values greater than 0 imply closed cohort sampling.
ar	numeric (scalar): Autocorrelation for exponential decay structure between periods within a cluster as described in Kasza et al (2019a;2019b). Available for cross-sectional sampling only (i.e. zeta=0 and iac=0). Default is ar=1 (no exponential decay). See details.
alpha	numeric (scalar): Statistical significance level. Default is 0.05.
retDATA	logical: if TRUE, all stored (input, intermediate, and output) values of swGlmPwr are returned. Default value is FALSE.
silent	logical: if TRUE, hides most warning messages. Default value is FALSE.

Details

Let Y_i denote a vector of observations for cluster i. The generalized linear mixed model formulation assumes

$$E(Y_i|b_i) = h(X\beta + Z_i b_i)$$

where h is a link function, β is a vector of fixed effects, b_i is a vector of random effects (in order, cluster, cluster*time, cluster*treatment) with $b_i \sim N(0, D)$ and X and Z are design matricies for the fixed and random effects, respectively. Then Breslow and Clayton (1993) show that

$$Var(\hat{\beta}) \approx \left(X^T V^{-1} X\right)^{-1}$$

where

$$V = W^b + ZDZ^T$$

where W^b is a diagonal matrix (W^b is approximated as W^0 in all calculations).

An exponentially decaying autocorrelation structure (only available for cross-sectional sampling i.e. zeta = 0) is incorporated by modifying V to

$$V = (W^b + ZD1Z^T) * R + ZD2Z^T$$

where $*$ denotes element-by-element matrix multiplication and, assuming there are random effects for cluster, cluster*time and cluster*treatment,

- $D1_{ij} = D$ for i and j <= J+1 (corresponding to the cluster and cluster*time random effects) and 0 otherwise

- $D2_{ij} = D$ for i or j > J+1 (corresponding to the cluster*treatment random effect) and 0 otherwise

- R = a toeplitz matrix with elements equal to ar^m where m = 0...J-1 is the distance from the main diagonal.

The two-sided statistical power for treatment effect θ (equal to H%*%fixed.treatment.effect if H is non-NULL) is

$$Pwr(\theta) = \Phi(\frac{Z - z_{1-\alpha/2}\sqrt{V_0(\hat{\theta})}}{\sqrt{V_\alpha(\hat{\theta})}}) + 1 - \Phi(\frac{Z + z_{1-\alpha/2}\sqrt{V_0(\hat{\theta})}}{\sqrt{V_\alpha(\hat{\theta})}})$$

, where Φ is the cumulative distribution function of the standard normal distribution.

The variance of $\hat{\theta}$ under the null is denoted as $V_0(\hat{\theta})$, and the variance of $\hat{\theta}$ under the alternative is denoted as $V_\alpha(\hat{\theta})$). Both variances are approximated by simplifying the Laplace approximation that marginalizes the random effects in the generalized linear mixed models. For more details, see Xia et al. (2020).

When the outcome is Gaussian, the method adopted by swGlmPwr coincides with that of swPwr, so power calculation for Gaussian outcomes is not included in swGlmPwr to avoid repetition. When the outcome is binomial, swGlmPwr performs power calculation on the natural scale (logit), while swPwr performs power calculation on the linear scale.

The value of zeta defines the samling scheme. When zeta = 0, cross-sectional sampling is assumed; if zeta > 0 then closed cohort sampling is assumed.

Incomplete designs (e.g. staircase) can be specified by using the swDsn function with the swBlk argument to specify the incomplete design or by specifying a complete design with swDsn and using the n argument of swPwr to specify a matrix with 0 in the missing cluster-periods. Note that if the second approach is used then H must be either a scalar or a vector as long as the longest lag in the complete design. See the examples below.

Value

numeric (scalar): swGlmPwr returns the power of treatment effect if retDATA = FALSE.

numeric (list): swGlmPwr returns all specified and computed items as objects of a list if retDATA = TRUE.

$design	list: A stepped wedge design object, typically from swDsn, that includes at least the following components: swDsn, swDsn.unique.clusters, clusters, n.clusters, total.time, nTxLev
$distn	character: Distribution assumed (binomial or Poisson). "binomial" implies binomial outcomes and "poisson" implies Poisson outcome.
$n	integer (scalar, vector, or matrix): Number of observations: (scalar) for all clusters and all time points; (vector) for each cluster at all time points; and (matrix) for each cluster at each time point, where rows correspond to clusters and columns correspond to time.

`$fixed.intercept`
: numeric (scalar): Intercept for the fixed effect on canonical scales (logit for binomial outcomes and log for Poisson outcomes).

`$fixed.treatment.effect`
: numeric (scalar): Coefficient(s) for the treatment(s) in the fixed effect model on canonical scales (logit for binomial outcomes and log for Poisson outcomes).

`$fixed.time.effect`
: numeric(scalar, vector): Coefficients for the time (as dummy variables) in the fixed effect model on canonical scales (logit for binomial outcomes and log for Poisson outcomes). The first time point is always used as reference. A common time effect for all time points after the first (scalar) or differnt time effects for each time point after the first (vector of length (total time-1)).

`H`
: numeric (vector): H specifies the desired linear combination of exposure time treatment effects for a ETI model-based estimate.

`$tau`
: numeric (scalar): Standard deviation of random intercepts on canonical scales (logit for binomial outcomes and log for Poisson outcomes).

`$eta`
: numeric (scalar): Standard deviation of random treatment effects on canonical scales (logit for binomial outcomes and log for Poisson outcomes).

`$rho`
: numeric (scalar): Correlation between random intercepts and random treatment effects on canonical scales (logit for binomial outcomes and log for Poisson outcomes).

`$gamma`
: numeric (scalar): Standard deviation of random time effects on canonical scales (logit for binomial outcomes and log for Poisson outcomes).

`$zeta`
: numeric (scalar): Standard deviation of individual random effects for closed cohort sampling on canonical scales (logit for binomial outcomes and log for Poisson outcomes).

`ar`
: numeric (scalar): Autocorrelation for exponential decay structure between periods within a cluster as described in Kasza et al (2019). Available for cross-sectional sampling only (i.e. zeta=0 and iac=0). See details.

`$alpha`
: numeric (scalar): Statistical significance level. Default is 0.05.

`$var.theta.null`
: numeric (martix): Variance-covariance martix of the estimated treatment effect(s) under the null for this SW CRT design.

`$var.theta.alt`
: numeric (marix): Variance-covariance matrix of the estimated treatment effect(s) under the alternative for this SW CRT design.

`$pwrGLM`
: numeric (scalar): Power of treatment effect using a simplified Laplace approximation.

Author(s)

Fan Xia, James P Hughes, and Emily C Voldal

References

Breslow NE and Clayton DG (1993). Approximate inference in generalized linear mixed models. Journal of the American Statistical Association, 88(421):9-25.

Kasza J, Hemming K, Hooper R, Matthews JNS, Forbes AB. Impact of non-uniform correlation on sample size and power in multiple-period cluster randomized trials. Statistical Methods in Medical Research 2019a; 28: 703-716.

Kasza J, Taljaard M, Forbes AB. Information content of stepped-wedge designs when treatment effect heterogeneity and/or implementation periods are present. Statistics in Medicine 2019b; 38:4686-4701.

Kenny A, Voldal E, Xia F, Heagerty PJ, Hughes JP. Analysis of stepped wedge cluster randomized trials in the presence of a time-varying treatment effect. Statistics in Medicine, in press, 2022.

Voldal EC, Hakhu NR, Xia, F, Heagerty PJ, Hughes JP. swCRTdesign: An R Package for Stepped Wedge Trial Design and Analysis. Computer Methods and Programs in Biomedicine 2020;196:105514.

Xia F, Hughes JP, Voldal EC, Heagerty PJ. Power and sample size calculation for stepped-wedge designs with discrete outcomes. Trials. 2021 Dec;22(1):598.

Examples

```
##test-case large clusters
logit <- function(x){log(x/(1 - x))}
#specify large cluster sizes
size = c(35219,53535,63785,456132,128670,96673,
51454,156667,127440,68615,56502,17719,75931,58655,52874,75936)

# Cross-sectional example
swGlmPwr(design=swDsn(c(4,3,5,4)),distn="binomial",n=size,
        fixed.intercept=log(28.62/(2*100000)),fixed.time.effect = 1,
        fixed.treatment.effect = log(.6),
        tau=.31,eta=abs(0.4*log(.6)),rho=0,gamma=.15,alpha=.05,retDATA = FALSE)

# Closed cohort example, comparing average of intervention period exposure
# lags 3 and 4 to control period
swGlmPwr(design=swDsn(c(5,5,5,5,5),extra.ctl.time=3,extra.trt.time=5),
         distn="binomial",n=20,
         fixed.intercept=logit(0.40),
         fixed.treatment.effect=c(0,0,rep(logit(0.60)-logit(0.40),8)),
         fixed.time.effect=0.08,
         H=c(0,0,1,1,0,0,0,0,0,0),
         tau=sqrt(.1316),gamma=sqrt(.1974),eta=0,zeta=sqrt(2.5))

# Example wih periods with no data and multiple treatment levels
stdy <- swDsn(c(6,6,6,6),swBlk=matrix(c(0,1,2,2,2,2,
                                        NA,0,1,2,2,2,
                                        NA,NA,0,1,2,2,
                                        NA,NA,NA,0,1,2),4,6,byrow=TRUE))
swGlmPwr(stdy, distn="binomial",n=10000,
         fixed.intercept=log(28.62/(2*100000)),fixed.time.effect = 1,
         fixed.treatment.effect = c(log(.6),log(.5)),
         tau=.31,eta=0,rho=0,gamma=0,alpha=.05,retDATA = FALSE)

#Incomplete design
# These all give the same power
design12 = swDsn(c(4,4,4),swBlk=matrix(c(0,1,1,NA,NA,NA,0,1,1,NA,NA,NA,0,1,1),3,5,byrow=TRUE))
```

```
swGlmPwr(design12,distn="binomial",n=10,fixed.intercept=0,fixed.treatment.effect=c(.8,.8),
         fixed.time.effect=0,H=c(.5,.5),tau=.1)
swGlmPwr(design12,distn="binomial",n=10,fixed.intercept=0,fixed.treatment.effect=c(.8,.8),
         fixed.time.effect=0,H=1,tau=.1) # H is automatically expanded and normalized
#
design12a = swDsn(c(4,4,4),extra.trt.time=1)
nmat12a=matrix(c(rep(c(10,10,10,0,0),4),rep(c(0,10,10,10,0),4),rep(c(0,0,10,10,10),4)),12,5,
               byrow=TRUE)
#Note maximum exposure time is 4 by design even though there are no data for exposure times 3 and 4
swGlmPwr(design12a,distn="binomial",n=nmat12a,fixed.intercept=0,
         fixed.treatment.effect=c(.8,.8,.8,.8),fixed.time.effect=0,H=1,tau=.1) #OK
swGlmPwr(design12a,distn="binomial",n=nmat12a,fixed.intercept=0,
      fixed.treatment.effect=c(.8,.8,.8,.8),fixed.time.effect=0,H=c(1,1,0,0),tau=.1) #OK
# The following generates an error; since design12a is a complete design with maximum lag 4,
# fixed.treatment.effect must be a vector length 4 and H must be a scalar or vector of length 4
# swGlmPwr(design12a,distn="binomial",n=nmat12a,fixed.intercept=0,fixed.treatment.effect=c(.8,.8),
#          fixed.time.effect=0,H=c(1,1),tau=.1)
```

---

swGlmSiz *Sample Size of Stepped Wedge Cluster Randomized Trial (SW CRT) using GLMM*

Description

swGlmSiz is a wrapper function for swGlmPwr. For a given design, number of clusters, power and fixed and random effect parameters, swGlmSiz computes the number of individuals per cluster needed to achieve the desired power.

Usage

```
swGlmSiz(power, alpha=0.05, ...)
```

Arguments

power	numeric (scalar) Desired power. Must be between α (default = 0.05) and 1.
alpha	numeric (scalar): Two-sided statistical significance level.
...	Arguments to be passed to swGlmPwr. See documentation for swGlmPwr.

Details

swGlmSiz calls swGlmPwr iteratively to find the number of participants per cluster-period needed to achieve power power. Note that all warnings and messages normally generated by swGlmPwr are turned off. Also, the argument retDATA is set to FALSE. It is recommended that after using swGlmSiz, the user call swGlmPwr with the recommended sample size to view any warning messages and/or to get the data normally returned when retDATA is TRUE.

An error in swGlmPwr causes swGlmSiz to terminate and the error message from swGlmPwr is printed.

Value

numeric (list): swGlmSiz returns a list of the following:

n	scalar: number of individuals per cluster-period required to achieve the desired power.
power	scalar: exact power, as computed by swGlmPwr, obtained with the sample size n.

Author(s)

Avi Kenny, James P Hughes

Examples

```
swGlmSiz(power = 0.8, design = swDsn(clusters=c(4,4,4,4)),distn = "binomial",
         fixed.intercept = log(0.1/0.9), fixed.treatment.effect = log(0.9),
         fixed.time.effect=0, tau = 0.2)
```

---

swPlot *Plot of Mean Response/Outcome for Stepped Wedge Cluster Randomized Trial (SW CRT)*

Description

swPlot returns plot of the mean response versus time based on waves and/or clusters from a SW CRT.

Usage

```
swPlot(response.var, tx.var, time.var, cluster.var, data, imputeNA = FALSE,
choose.mfrow=NULL, by.wave=TRUE, combined.plot=TRUE, choose.xlab="Time", choose.main=NULL,
choose.pch=NULL, choose.cex=1, choose.tx.col=NULL, choose.tx.lty = c(2,1),
choose.ncol=2, xlim=NULL, ylim=NULL, choose.tx.pos="topright", choose.legend.pos="right")
```

Arguments

response.var	numeric(vector): Response (Outcome) variable.
tx.var	numeric(vector): Treatment (Predictor of Interest) variable. Typically, 0=control, 1=intervention, and values greater than 1 correspond to other treatment levels.
time.var	integer(vector): Time (points) variable, corresponding to the time points when data were collected during the SW CRT.
cluster.var	integer(vector) or character(vector): Cluster (identification) variable, corresponding to the cluster where an observation is from.
data	An optional data frame containing (at least) the response, treatment (tx), time, and cluster variables.

imputeNA	logical (scalar): If TRUE, then swPlot (via a call to swSummary) attempts to impute the intervention status for cluster-periods with 0 non-missing observations (see Details). If FALSE, cluster-periods with 0 non-missing observations are assigned NA. Note that changing imputeNA from FALSE to TRUE may also change the inferred number of unique waves. Default is FALSE.
choose.mfrow	numeric (vector): Choose mfrow for plot. If NULL, mfrow is automatically assigned based on the plot created. The default is NULL.
by.wave	logical: If TRUE, plot mean response according to each wave. If FALSE, plot mean response according to each cluster. The default is TRUE.
combined.plot	logical: If TRUE, plot mean response on same plot (what is plotted depends on by.wave). If FALSE, plot mean response on separate plots for each wave (what is plotted depends on by.wave). The default is TRUE.
choose.xlab	Choose xlab for plot. The default is "Time".
choose.main	Choose main for plot. If NULL, main is chosen for the user; which is highly recommended. The default is NULL.
choose.pch	Choose pch for plot. If NULL, pch are chosen for the user; which is highly recommended. The default is NULL.
choose.cex	Choose cex for choose.pch. Standard cex option in points() applies. The default is 1.
choose.tx.col	Choose colors for different treatment options. Vector of two colors, corresponding to control and treatment groups, respectively. If NULL, colors are chosen for the unique treatment options in the data supplied. The default is NULL.
choose.tx.lty	Choose line types for different treatment options. Vector of two numbers for lty, corresponding to control and treatment groups, respectively. The default is c(2,1).
choose.ncol	Choose number of columns for non-treatment legend. Standard ncol option in legend() applies. The default is 2.
xlim, ylim	Vectors of length 2 giving x and y axis limits on plot. If NULL, minimum and maximum of summary time and response variables are used, respectively.
choose.tx.pos	Choose where to place treatment colors "legend". Standard legend() positions apply. The default is "topright"; if this covers points, it should be changed.
choose.legend.pos	Choose where to place the non-treatment legend. Standard legend() positions apply. The default is "right"; if this covers points, it should be changed. If "mouseclick", user specifies location with mouse/trackpad by clicking on the plot that appears. Specify NULL to remove legend. When by.wave=FALSE, no legend is plotted

Details

If imputeNA is set to TRUE then any NA in the design matrix is set to 1 (intervention) if the NA follows a 1 in the same cluster; is set to 0 (control) if the NA precedes a 0 in the same cluster; and is otherwise left as NA.

Returns a plot of the mean response versus time with a combination of by wave (TRUE / FALSE) and combined plot (TRUE / FALSE) from a SW CRT. See discussion of number of waves in help file for swSummary.

Author(s)

James P Hughes, Navneet R Hakhu, and Emily C Voldal

References

Hussey MA, Hughes JP. Design and analysis of stepped wedge cluster randomized trials. Contemporary Clinical Trials 2007;28:182-191.

Examples

```
library(swCRTdesign)
# Example 1
# Generate binary response with 5 clusters, 4 time points
data.Ex1 <- swSim(swDsn(c(2,2,1)),family=binomial(link="identity"), n=120,
                    mu0=0.25, mu1=0.30,time.effect=0, tau=0.05, gamma=.01)

# Example 1 (Mean Response vs Time, by.wave=TRUE, combined.plot=TRUE)
swPlot(response.var, tx.var, time.var, cluster.var, data.Ex1, by.wave=TRUE,
combined.plot=TRUE, choose.tx.pos="bottomright", choose.legend.pos="bottom")

# Example 2 (Mean Response vs Time, by.wave=TRUE, combined.plot=FALSE)
swPlot(response.var, tx.var, time.var, cluster.var, data.Ex1, by.wave=TRUE,
combined.plot=FALSE, choose.tx.pos="bottomright", choose.legend.pos="bottom")

# Example 3 (Mean Response vs Time, by.wave=FALSE, combined.plot=TRUE)
swPlot(response.var, tx.var, time.var, cluster.var, data.Ex1, by.wave=FALSE,
combined.plot=TRUE, ylim=c(0,0.6), choose.tx.pos="bottomright", choose.legend.pos="topleft")

# Example 4 (Mean Response vs Time, by.wave=FALSE, combined.plot=FALSE)
swPlot(response.var, tx.var, time.var, cluster.var, data.Ex1, by.wave=FALSE,
combined.plot=FALSE, choose.tx.pos="bottomright", choose.legend.pos="bottom")
```

`swPwr` *Power of Stepped Wedge Cluster Randomized Trial (SW CRT)*

Description

`swPwr` returns (two-sided) power for the treatment effect(s) from an immediate treatment effect (IT) model or an exposure time indicator (ETI) model (Kenny et al, 2022) for the specified SW CRT design using a linear models weighted least squares (WLS) approach. A cross-sectional or closed cohort sampling scheme can be specified. The response/outcome of interest can be binomial or Gaussian distributed and is assumed to come from a model with random intercepts, random treatment effects, random cluster-specific time effects, and, in the case of closed cohort sampling, an individual random effect. Variance components can be specified using either random effect variances (tau, eta, rho, gamma, and zeta) or icc, cac, iac (see details). If a cross-sectional sampling, random intercepts only model is used (i.e., eta, rho, gamma and zeta are 0 and n is constant over clusters and time), then the power calculation is comparable to the closed-form formula of [Hussey and Hughes, 2007]. See [Voldal et al., 2020] for more guidance. This function is also available as a Shiny app at https://swcrtdesign.shinyapps.io/stepped_wedge_power_calculation/.

Usage

```
swPwr(design, distn, n, mu0, mu1, H=NULL, sigma, tau=0, eta=0, rho=0, gamma=0, zeta=0,
icc=0, cac=1, iac=0, ar=1, alpha=0.05, retDATA=FALSE, silent=FALSE)
```

Arguments

design	list: A stepped wedge design object, typically from swDsn, that includes at least the following components: swDsn, swDsn.unique.clusters, clusters, n.clusters, total.time, nTxLev, TxLev. Fractional treatment effects specified in swDsn are used in the calculation.
distn	character: Distribution assumed (gaussian or binomial). Currently, 'Binomial' implies Bernoulli.
n	integer (scalar, vector, or matrix): Number of observations: (scalar) for all clusters and all time points that are not NA in design$swDsn; (vector) for each cluster at all time points that are not NA in design$swDsn; and (matrix) for each cluster at each time point, where rows correspond to clusters and columns correspond to time. n can also be used to specify a design with transition periods (a transition period is a period, typically, immediately after treatment is introduced were no data are collected). Simply define n as a matrix with a sample size of 0 during every transition period. This is equivalent to putting an NA in the corresponding cluster-period in design$swDsn.
mu0	numeric (scalar): Mean outcome in the control group. See also documentation for H, below.
mu1	numeric (scalar or vector): Mean outcome in the treatment group(s). The treatment effect is the difference in means $\theta = \mu_1 - \mu_0$. If tx.effect.frac is specified in the call to swDsn then the assumed treatment effect at exposure time k is tx.effect.frac[k]*θ. For an IT model with multiple treatment levels the length of mu1 should be equal to the number of treatment levels. See also documentation for H, below.
H	numeric (vector): If NULL, then swPwr assumes an immediate, constant treatment effect (IT) model. If not NULL, then an exposure time indicator (ETI) model is assumed and H should be a vector as long as the longest exposure time (in a classic stepped wedge design, the longest exposure time is the number of time periods minus one). H specifies the desired linear combination of exposure time treatment effects. For example, in a stepped wedge trial with 5 time periods and four exposure times, H = rep(.25,4) gives the average treatment effect over the four exposure times; H = c(0,0,.5,.5) ignores the first two periods after the intervention is introduced and averages the remaining periods. Typically, the sum of H is 1.0; if not, it is renormalized to sum to 1.0. H can only be specified when there is a single intervention type (i.e. design$swDsn includes only NA,0,1; see cautions in help for swDsn about including NA periods when H is used). mu0 and mu1 give expected values of the linear combination of exposure time treatment effects under the null and alternative hypotheses, respectively. Do not use an ETI model (non-null H) if you specified a fractional treatment effect in swDsn. See the details section for additional notes regarding incomplete designs.

`sigma` numeric (scalar): Standard deviation when assuming Gaussian distribution (`distn=gaussian`). For binomial distribution σ^2 is automatically set to $\bar{\mu}(1-\bar{\mu})$ where $\bar{\mu} = (\mu_1 + \mu_0)/2$

`tau` numeric (scalar): Standard deviation of random intercepts.

`eta` numeric (scalar or list): Standard deviation of random treatment effect(s). If there is only one treatment group or if all the treatment groups have identical standard deviations for their random treatment effects, then eta can be specified as a scalar. If there are multiple treatment effects then eta can be specified as a list with two components: eta$eta is a vector of standard deviations of the random treatment effects (must be of length design$nTxLev); eta$corr is the correlation matrix of the random treatment effects (must be square and symmetrical with dimension design$nTxLev). If there are, say, two treatment groups then specifying eta = .005 is equivalent to specifying eta = list(eta=c(.005,.005),corr=matrix(1,2,2)).

`rho` numeric (scalar or vector): Correlation between random intercepts and random treatment effects. If there are multiple treatment effects and eta has been specified as a list, then rho should have the same length as eta$eta.

`gamma` numeric (scalar): Standard deviation of random time effects.

`zeta` numeric (scalar): Standard deviation of individual effects for closed cohort sampling. Default is 0 which implies cross-sectional sampling. Values greater than 0 imply closed cohort sampling.

`icc` numeric (scalar): Within-period intra-cluster correlation. Can be used with CAC instead of tau, eta, rho, and gamma; see details.

`cac` numeric (scalar): Cluster auto-correlation. Can be used with ICC instead of tau, eta, rho, and gamma; see details.

`iac` numeric (scalar): Individual auto-correlation for closed cohort sampling. Can be used with ICC and CAC instead of tau, eta, rho, gamma, and zeta; see details. Default is 0 which implies cross-sectional sampling. Values greater than 0 imply closed cohort sampling.

`ar` numeric (scalar): Autocorrelation for exponential decay structure between periods within a cluster as described in Kasza et al (2019a;2019b). Available for cross-sectional sampling only (i.e. zeta=0 and iac=0). Default is ar=1 (no exponential decay). See details.

`alpha` numeric (scalar): Two-sided statistical significance level.

`retDATA` logical: if TRUE, all stored (input, intermediate, and output) values of `swPwr` are returned. Default value is `FALSE`.

`silent` logical: if TRUE, hides most warning messages. Default value is `FALSE`.

Details

If ar=1, we assume the outcome variable of interest can be modeled using random effects

$$Y_{ijk} = \mu_0 + \beta_j + \theta X_{ij} + a_i + b_{ij} + r_i X_{ij} + c_{ik} + e_{ijk}$$

where

$$a_i \sim N(0, \tau^2)$$

$$b_{ij} \sim N(0, \gamma^2)$$
$$r_i \sim N(0, \eta^2)$$
$$Cov(a_i, r_i) = \rho\tau\eta$$
$$c_{ik} \sim N(0, \zeta^2)$$
$$e_{ijk} \sim N(0, \sigma^2)$$

for K observations in cluster $i = 1, \ldots, I$ and at time $j = 1, \ldots, J$.

An exponentially decaying autocorrelation structure is specified by setting ar<1. The exponential decay only applies to the cluster and cluster*time random effects (Kasza et al., 2019) and is only available for cross-sectional sampling (i.e. `zeta = iac = 0`). Let Y_i represent the vector of cluster-period means for cluster i. Then the covariance of Y_i is

$$Var(Y_i) = G * R + T$$

where $*$ denotes element by element multiplication and G, R and T are JxJ matricies with

$$\begin{array}{rcl} G & = & \tau^2 + diag(\gamma^2) + diag(\sigma^2/K) \\ T_{ij} & = & \eta^2 \quad \text{for i=j=J} \\ & = & \rho\tau\eta \quad \text{for i=1...J-1, j=J and i=J, j=1...J-1} \\ & = & 0 \quad \text{elsewhere} \\ R & = & \text{a toeplitz matrix with elements equal to } \mathtt{ar}^m \text{ where m = 0...J-1 is the distance from the main diagonal} \end{array}$$

The two-sided statistical power of treatment effect ($\theta = \mu_1 - \mu_0$) is

$$Pwr(\theta) = \Phi(Z - z_{1-\alpha/2}) + \Phi(-Z - z_{1-\alpha/2})$$

where

$$Z = \frac{|\theta|}{\sqrt{Var(\hat{\theta}_{WLS})}}$$

and Φ is the cumulative distribution function of the standard normal $N(0, 1)$ distribution. If H is non-NULL then the μ are assumed to be equal to Hδ where δ is a vector of exposure time treatment effects.

The value of zeta defines the samling scheme. When zeta = 0, cross-sectional sampling is assumed; if zeta > 0 then closed cohort sampling is assumed.

When eta (and rho) are 0, instead of using tau, eta, rho, gamma and zeta, the icc, cac and iac can be used to define the variability of the random effects. In this model,

$$ICC = \frac{\tau^2 + \gamma^2}{\tau^2 + \gamma^2 + \zeta^2 + \sigma^2}$$

$$CAC = \frac{\tau^2}{\tau^2 + \gamma^2}$$

$$IAC = \frac{\zeta^2}{\zeta^2 + \sigma^2}$$

Choose one parameterization or the other, do not mix parameterizations.

Incomplete designs (e.g. staircase) can be specified by using the swDsn function with the swBlk argument to specify the incomplete design or by specifying a complete design with swDsn and using the n argument of swPwr to specify a matrix with 0 in the missing cluster-periods. Note that if the second approach is used then H must be either a scalar or a vector as long as the longest lag in the complete design. See the examples below.

Value

numeric (vector): swPwr returns the power of treatment effect(s), where the variance of treatment effect is computed by WLS.

numeric (list): swPwr returns all specified and computed items as objects of a list if retDATA = TRUE.

design	list: The stepped wedge design object as input.
distn	character: Distribution assumed (gaussian or binomial).
n	integer (scalar, vector, or matrix): Number of observations: (scalar) for all clusters and all time points; (vector) for each cluster at all time points; and (matrix) for each cluster at each time point, where rows correspond to clusters and columns correspond to time.
mu0	numeric (scalar): Mean outcome in the control group.
mu1	numeric (scalar): Mean outcome in intervention group(s). Note: treatment effect is difference in means $\theta = \mu_1 - \mu_0$.
H	numeric (vector): H specifies the desired linear combination of exposure time treatment effects for a ETI model-based estimate.
sigma	numeric (scalar): Standard deviation input. For binomial distribution, sigma = NA
tau	numeric (scalar): Standard deviation of random intercepts.
eta	numeric (scalar): Standard deviation of random treatment effects.
rho	numeric (scalar): Correlation between random intercepts and random treatment effects.
gamma	numeric (scalar): Standard deviation of random time effects.
zeta	numeric (scalar): Standard deviation of individual random effects for closed cohort sampling.
ar	numeric (scalar): Autocorrelation for exponential decay structure between periods within a cluster as described in Kasza et al (2019). Available for cross-sectional sampling only (i.e. zeta=0 and iac=0). See details.
alpha	numeric (scalar): Statistical significance level.
Xmat	numeric (matrix): Design matrix for this SW CRT design.
Wmat	numeric (matrix): Covariance matrix for this SW CRT design.
var	Variance-covariance matrix of treatment effect(s). Can be used to calculate power for constrasts other than control versus treatment
var.theta.WLS	numeric (scalar): Variance(s) of treatment effect(s) using weighted least squares (WLS) for this SW CRT design.
pwrWLS	numeric (scalar): Power of treatment effect (θ) using weighted least squares (WLS) for this SW CRT design.

Author(s)

James P Hughes, Navneet R Hakhu, and Emily C Voldal

References

Hussey MA, Hughes JP. Design and analysis of stepped wedge cluster randomized trials. Contemporary Clinical Trials 2007;28:182-191.

Kasza J, Hemming K, Hooper R, Matthews JNS, Forbes AB. Impact of non-uniform correlation on sample size and power in multiple-period cluster randomized trials. Statistical Methods in Medical Research 2019a; 28: 703-716.

Kasza J, Taljaard M, Forbes AB. Information content of stepped-wedge designs when treatment effect heterogeneity and/or implementation periods are present. Statistics in Medicine 2019b; 38:4686-4701.

Kenny A, Voldal E, Xia F, Heagerty PJ, Hughes JP. Analysis of stepped wedge cluster randomized trials in the presence of a time-varying treatment effect. Statistics in Medicine, in press, 2022.

Voldal EC, Hakhu NR, Xia F, Heagerty PJ, Hughes JP. swCRTdesign: An R package for stepped wedge trial design and analysis. Computer Methods and Programs in Biomedicine 2020;196:105514.

Examples

```
# Example 1 (Random Intercepts Only, standard Stepped Wedge (SW) design)
swPwr.Ex1.RIO.std <- swPwr(swDsn(c(6,6,6,6)), distn="binomial",
n=120, mu0=0.05, mu1=0.035, tau=0.01, eta=0, rho=0, gamma=0, alpha=0.05, retDATA=FALSE)
swPwr.Ex1.RIO.std

# Example 2 (Random Intercepts Only, extended SW design)
swPwr.Ex1.RIO.extend <- swPwr(swDsn(c(6,6,6,6), extra.trt.time=3), distn="binomial",
n=120, mu0=0.05, mu1=0.035, tau=0.01, eta=0, rho=0, gamma=0,
alpha=0.05, retDATA=FALSE)
swPwr.Ex1.RIO.extend

# Example 3 (Independent Random Intercepts and Treatment effects, standard SW design)
swPwr.Ex1.IRIS <- swPwr(swDsn(c(6,6,6,6)), distn="binomial",
n=120, mu0=0.05, mu1=0.035, tau=0.01, eta=0.0045, rho=0, gamma=0,
alpha=0.05, retDATA=FALSE)
swPwr.Ex1.IRIS

# Example 4 (Correlated Random Intercepts and Slopes, standard SW design)
swPwr.Ex1.CRIS <- swPwr(swDsn(c(6,6,6,6)), distn="binomial",
n=120, mu0=0.05, mu1=0.035, tau=0.01, eta=0.0045, rho=0.4, gamma=0,
alpha=0.05, retDATA=FALSE)
swPwr.Ex1.CRIS

# Example 5 (Random time effect and correlated Random Intercepts and Slopes, standard SW design)
swPwr.Ex1.RTCRIS <- swPwr(swDsn(c(6,6,6,6)), distn="binomial",
n=120, mu0=0.05, mu1=0.035, tau=0.01, eta=0.0045, rho=0.4, gamma = 0.1,
alpha=0.05, retDATA=FALSE)
swPwr.Ex1.RTCRIS

#Example 6 (Sample size varying by cluster)
```

```
sample.size.vector <- c(35219,53535,63785,456132,128670,96673,
           51454,156667,127440,68615,56502,17719,
           75931,58655,52874,75936)
swPwr.Ex1.vector <- swPwr(swDsn(c(4,3,5,4)), distn="gaussian",
n=sample.size.vector, mu0=2.66, mu1=2.15,
sigma=sqrt(1/2.66), tau=0.31, eta=0.2, rho=0, gamma = 0.15,
alpha=0.05, retDATA=FALSE)
swPwr.Ex1.vector

#Example 7 (Sample size varying by cluster and time)
sample.size.matrix <- matrix(c(26, 493,  64,  45,  48, 231, 117,  17,  49,  36,  19,  77, 67, 590,
261, 212,  67, 318, 132,  58,  44,  57,  59,  78, 115, 532, 176, 199,  73, 293, 129,  79,  51,
62, 109,  94, 174, 785, 133,  79, 120, 305, 224,  99,  83,  79, 122, 122, 94, 961,  90, 131, 166,
352, 316,  59,  54, 131, 101, 133),nrow=12,ncol=5, byrow=FALSE)
swPwr.Ex1.matrix <- swPwr(swDsn(c(3,3,3,3)), distn="binomial",
n=sample.size.matrix, mu0=0.08, mu1=0.06, tau=0.017, eta=0.006, rho=-0.5, gamma = 0,
alpha=0.05, retDATA=FALSE)
swPwr.Ex1.matrix

#Example 8 (Using ICC and CAC)
swPwr.Ex1.icccac <- swPwr(swDsn(c(6,6,6,6)), distn="gaussian",
n=120, mu0=0.05, mu1=0.035, sigma=0.1, icc=0.02, cac=0.125, alpha=0.05, retDATA=FALSE)
swPwr.Ex1.icccac

# Example 9 (Random time effect, closed cohort sampling)
sample_size = matrix(c(rep(c(20,20,20,20,20,20,20,20,20,20,20,20,20,20),5),
                       rep(c( 0,20,20,20,20,20,20,20,20,20,20,20,20,20),5),
                       rep(c( 0, 0,20,20,20,20,20,20,20,20,20,20,20,20),5),
                       rep(c( 0, 0, 0,20,20,20,20,20,20,20,20,20,20,20),5),
                       rep(c( 0, 0, 0, 0,20,20,20,20,20,20,20,20,20,20),5)),
                       25,14,byrow=TRUE)
swPwr.Ex9 <- swPwr(design=swDsn(c(5,5,5,5,5),extra.ctl.time=3,extra.trt.time=5),
      distn="gaussian",n=sample_size,mu0=-0.4,mu1=0.4,
      sigma=1.0,tau=sqrt(.1316),gamma=sqrt(.1974),eta=0,rho=0,zeta=sqrt(2.5),
      silent=TRUE)
swPwr.Ex9

#Example 10 (Periods with no data, multiple treatment levels)
stdy <- swDsn(c(6,6,6,6),swBlk=matrix(c(0,1,2,2,2,2,
                                        NA,0,1,2,2,2,
                                        NA,NA,0,1,2,2,
                                        NA,NA,NA,0,1,2),4,6,byrow=TRUE))
swPwr.Ex10 <-swPwr(stdy, distn="binomial",n=120, mu0=0.05, mu1=c(0.035,0.03),
      tau=0.01, eta=0, rho=0, gamma=0, alpha=0.05, retDATA=TRUE, silent=TRUE)
swPwr.Ex10

# EXample 11 Compare to Kasza et al (2019) figure 4, lower left
swPwr(swDsn(c(1,1,1,1,1,1,1)),distn="gaussian",n=50,mu0=0,mu1=0.2,
          tau=sqrt(.035),sigma=sqrt(1-.035),ar=.95,alpha=0.05,retDATA=FALSE,silent=TRUE)

# Example 12 Incomplete design
# These all give the same power
design12 = swDsn(c(4,4,4),swBlk=matrix(c(0,1,1,NA,NA,NA,0,1,1,NA,NA,NA,0,1,1),3,5,byrow=TRUE))
```

```
swPwr(design12,distn="gaussian",n=10,mu0=0,mu1=.5,H=c(.5,.5),sigma=1,icc=.01,silent=TRUE)
swPwr(design12,distn="gaussian",n=10,mu0=0,mu1=.5,H=1,sigma=1,icc=.01,silent=TRUE)
# H is automatically expanded and normalized
#
design12a = swDsn(c(4,4,4),extra.trt.time=1)
nmat12a=matrix(c(rep(c(10,10,10,0,0),4),rep(c(0,10,10,10,0),4),rep(c(0,0,10,10,10),4)),12,5,
               byrow=TRUE)
swPwr(design12a,distn="gaussian",n=nmat12a,mu0=0,mu1=.5,H=1,sigma=1,icc=.01,silent=TRUE) #OK
swPwr(design12a,distn="gaussian",n=nmat12a,mu0=0,mu1=.5,H=c(1,1,0,0),sigma=1,icc=.01,
      silent=TRUE) #OK
# The following generates an error; since design12a is a complete design with maximum lag 4,
# H must be a scalar or vector of length 4
# swPwr(design12a,distn="gaussian",n=nmat12a,mu0=0,mu1=.5,H=c(1,1),sigma=1,icc=.01,silent=TRUE)
```

`swSim` *Simulating individual-level data for specified study design of Stepped Wedge Cluster Randomized Trial (SW CRT)*

Description

`swSim` returns individual-level data set of a SW CRT study design for the specified number of clusters per wave, treatment effect (possibly varying according to the exposure time - time after crossing over from control), time effect, family (and link function), number of individuals per cluster per time period, mean in control arm, mean in treatment arm(s), standard deviation (if applicable), standard deviation of random intercepts, standard deviation of random treatment effects, correlation between random intercepts and random treatment effects, standard deviation of random time effects, and, for closed cohorts, standard deviation of individual random effects. Alternatively, for a Gaussian family, standard deviations of random effects may be specified using ICC, CAC and, for closed cohorts, IAC; see `swPwr` details. An option to add time point labels is also included.

Usage

```
swSim(design, family, log.gaussian = FALSE, n, mu0, mu1, time.effect,
sigma, tau=0, eta=0, rho=0, gamma=0, zeta=0, icc=0, cac=1, iac=0,
time.lab = NULL, retTimeOnTx=TRUE, silent = FALSE, nocheck=FALSE)
```

Arguments

`design`	list: A stepped wedge design object, typically from `swDsn`, that includes at least the following components: swDsn, swDsn.unique.clusters, clusters, n.clusters, total.time, nTxLev, TxLev. Fractional treatment effects specified in `swDsn` are used in the simulation.
`family`	character: Used in typical way. However, only Gaussian, Binomial, and Poisson families accepted. Also, only identity, logit, and log links accepted. Logit link is only available for Binomial family, and log link is only available for Binomial

and Poisson. Currently, 'Binomial' implies Bernoulli. Default links are identity for Gaussian, logit for Binomial, log for Posson. ***NOTE: It is the users responsibility to make sure specified parameters (mu0, mu1, time.effect, tau, eta, rho, gamma) are ALL on the SAME scale as the specified link function; see example.***

`log.gaussian` character: When `TRUE` with a Gaussian family, simulates data whose log follows a Gaussian distribution; all parameters (mu0, mu1, time.effect, variance parameters) refer to the log scale. Default is `FALSE`.

`n` integer (scalar, vector, or matrix): Number of observations: (scalar) for all clusters and all time points; (vector) for each cluster at all time points; and (matrix) for each cluster at each time point, where rows correspond to clusters and columns correspond to time. `n` can also be used to specify a design with transition periods were no data is collected; see `swPwr`.

`mu0` numeric (scalar): Mean outcome in the control group on the appropriate scale.

`mu1` numeric (scalar or vector): Mean outcome in the treatment group on the appropriate scale. If design$nTxLev > 1 (number of treatment levels greater than 1) then `mu1` should have length equal to `design$nTxLev`; if `design$nTxLev = 1` then `mu1` should be either a scalar (constant treatment effect) or a vector with length equal to the maximum number of exposure times (time-varying treatment effect). Do not specify time-varying or multiple treatment effects if the design includes fractional treatment effects.

`time.effect` integer (scalar or vector): Time effect at each time point on the appropriate scale (added to mean at each time).

`sigma` numeric (scalar): Pooled treatment and control arm standard deviation on the appropriate scale. Ignored if family != Gaussian.

`tau` numeric (scalar): Standard deviation of random intercepts on the appropriate scale. Default is 0.

`eta` numeric (scalar): Standard deviation of random treatment effects on the appropriate scale. Default is 0.

`rho` numeric (scalar): Correlation between random intercepts and random treatment effects on the appropriate scale. Default is 0.

`gamma` numeric (scalar): Standard deviation of random time effects on the appropriate scale. Default is 0.

`zeta` numeric (scalar): Standard deviation of individual effects for closed cohort sampling on the appropriate scale. Default is 0 which implies cross-sectional sampling. Values greater than 0 imply closed cohort sampling.

`icc` numeric (scalar): Within-period intra-cluster correlation on the appropriate scale. Can be used with CAC instead of tau, eta, rho, and gamma when the outcome is Gaussian. Default is 0.

`cac` numeric (scalar): Cluster auto-correlation on the appropriate scale. Can be used with ICC instead of tau, eta, rho, and gamma when the outcome is Gaussian. Default is 1.

`iac` numeric (scalar): Individual auto-correlation for closed cohort sampling on the appropriate scale. Can be used with ICC and CAC instead of tau, eta, rho, gamma, and zeta when the outcome is gaussian. Default is 0 which implies cross-sectional sampling. Values greater than 0 imply closed cohort sampling.

time.lab	character (vector): Labeling for time points when output is display; choice of labeling does not affect results.
retTimeOnTx	This argument has been deprecated. TimeOnTx is always returned.
silent	logical: if TRUE, hides reminder about order of entries in n when n is not a scalar. Default value is FALSE.
nocheck	logical: if TRUE, various checks on the validity of the arguments are not done. For internal use only.

Details

When simulating from a Gaussian distribution and if eta (and rho) are 0, instead of using tau, eta, rho, gamma and zeta, the icc, cac and iac can be used to define the variability of the random effects. In this model,

$$ICC = \frac{\tau^2 + \gamma^2}{\tau^2 + \gamma^2 + \zeta^2 + \sigma^2}$$

$$CAC = \frac{\tau^2}{\tau^2 + \gamma^2}$$

$$IAC = \frac{\zeta^2}{\zeta^2 + \sigma^2}$$

Choose one parameterization or the other, do not mix parameterizations.

Value

numeric (data frame): Returns the following (individual-level) variables corresponding to the specified SW CRT design:

$response.var	numeric (vector): Response variable based on specified SW CRT design of interest (including family and link function) for each observation in the data frame/set.
$tx.var	numeric (vector): Predictor of interest. (Fractional) treatment effect corresponding to 0=control, 1=treatment, and value between 0 and 1 corresponding to treatment arm with fractional treatment effect (for each observation in the data frame/set).
$timeOnTx.var	numeric (vector): Predictor of interest when interested in time on treatment lag effect. Total time spent on treatment for each observation in the data frame/set, with 0=control, 1=first time period on treatment, 2=second time period on treatment, etc.
$time.var	numeric (vector): Time point id for each observation in the data frame/set.
$cluster.var	numeric (vector): Grouping variable. Cluster id for each observation in the data frame/set.

Author(s)

James P Hughes, Navneet R Hakhu, and Emily C Voldal

References

Hussey MA, Hughes JP. Design and analysis of stepped wedge cluster randomized trials. Contemporary Clinical Trials 2007;28:182-191.

Voldal EC, Hakhu NR, Xia, F, Heagerty PJ, Hughes JP. swCRTdesign: An R Package for Stepped Wedge Trial Design and Analysis. Computer Methods and Programs in Biomedicine 2020;196:105514.

Examples

```
library(swCRTdesign)
# Example 1 [ n = scalar; can be vector (for different n for each cluster,
# n=rep(120,22)) or matrix (different n for each cluster at each time point,
# n=matrix(120,22,5)) ]

# generate cross-sectional SW data
design <- swDsn(clusters=c(6,6,6,4))
set.seed(5)
swGenData.nScalar <- swSim( design, family=binomial(link="logit"), n=120,
mu0=log(0.1/0.9), mu1=log(0.9) + log(0.1/0.9), time.effect=0, tau=0.2, eta=0,
rho=0, gamma=0, time.lab=seq(0,12,3))

# summarize SW data by wave
swSummary(response.var, tx.var, time.var, cluster.var, swGenData.nScalar,
type="mean", digits=3)$response.wave

swSummary(response.var, tx.var, time.var, cluster.var, swGenData.nScalar,
type="mean", digits=3)$swDsn
```

---

swSim2 *Simulating individual-level data for specified study design of Stepped Wedge Cluster Randomized Trial (SW CRT)*

Description

swSim2 is similar to swSim in that it returns individual-level data set of a SW CRT study design for the specified number of clusters per wave, treatment effect (possibly varying according to the exposure time - time after crossing over from control), time effect, family (and link function), number of individuals per cluster per time period, mean in control arm, mean in treatment arm(s), standard deviation (if applicable), standard deviation of random intercepts, standard deviation of random treatment effects, correlation between random intercepts and random treatment effects, standard deviation of random time effects, and, for closed cohorts, standard deviation of individual random effects. Alternatively, for a Gaussian family, standard deviations of random effects may be specified using ICC, CAC and, for closed cohorts, IAC; see swPwr details. An option to add time point labels is also included.

swSim2 extends the functionality of swSim by allowing one to simulate data with an exponentially decaying autocorrelation structure as described in Kasza et al. (2019). swSim2 uses the glmmTMB function simulate_new to simulate datasets. swSim has been retained for backwards compatability.

Usage

```
swSim2(design, family, log.gaussian = FALSE, n, mu0, mu1, time.effect,
sigma, tau=0, eta=0, rho=0, gamma=0, zeta=0, icc=0, cac=1, iac=0, ar=1,
time.lab = NULL, silent = FALSE, nocheck=FALSE)
```

Arguments

design	list: A stepped wedge design object, typically from swDsn, that includes at least the following components: swDsn, swDsn.unique.clusters, clusters, n.clusters, total.time, nTxLev, TxLev. Fractional treatment effects specified in swDsn are used in the simulation.
family	character: Used in typical way. However, only Gaussian, Binomial, and Poisson families accepted. Also, only identity, logit, and log links accepted. Logit link is only available for Binomial family, and log link is only available for Binomial and Poisson. Currently, 'Binomial' implies Bernoulli. Default links are identity for Gaussian, logit for Binomial, log for Posson. ***NOTE: It is the users responsibility to make sure specified parameters (mu0, mu1, time.effect, tau, eta, rho, gamma) are ALL on the SAME scale as the specified link function; see example.***
log.gaussian	character: When TRUE with a Gaussian family, simulates data whose log follows a Gaussian distribution; all parameters (mu0, mu1, time.effect, variance parameters) refer to the log scale. Default is FALSE.
n	integer (scalar, vector, or matrix): Number of observations: (scalar) for all clusters and all time points; (vector) for each cluster at all time points; and (matrix) for each cluster at each time point, where rows correspond to clusters and columns correspond to time. n can also be used to specify a design with transition periods were no data is collected; see swPwr.
mu0	numeric (scalar): Mean outcome in the control group on the appropriate scale.
mu1	numeric (scalar or vector): Mean outcome in the treatment group on the appropriate scale. If design$nTxLev > 1 (number of treatment levels greater than 1) then mu1 should have length equal to design$nTxLev; if design$nTxLev = 1 then mu1 should be either a scalar (constant treatment effect) or a vector with length equal to the maximum number of exposure times (time-varying treatment effect). Do not specify time-varying or multiple treatment effects if the design includes fractional treatment effects.
time.effect	integer (scalar or vector): Time effect at each time point on the appropriate scale (added to mean at each time).
sigma	numeric (scalar): Pooled treatment and control arm standard deviation on the appropriate scale. Ignored if family != Gaussian.
tau	numeric (scalar): Standard deviation of random intercepts on the appropriate scale. Default is 0.
eta	numeric (scalar): Standard deviation of random treatment effects on the appropriate scale. Default is 0.
rho	numeric (scalar): Correlation between random intercepts and random treatment effects on the appropriate scale. Default is 0.

gamma	numeric (scalar): Standard deviation of random time effects on the appropriate scale. Default is 0.
zeta	numeric (scalar): Standard deviation of individual effects for closed cohort sampling on the appropriate scale. Default is 0 which implies cross-sectional sampling. Values greater than 0 imply closed cohort sampling.
icc	numeric (scalar): Within-period intra-cluster correlation on the appropriate scale. Can be used with CAC instead of tau, eta, rho, and gamma when the outcome is Gaussian. Default is 0.
cac	numeric (scalar): Cluster auto-correlation on the appropriate scale. Can be used with ICC instead of tau, eta, rho, and gamma when the outcome is Gaussian. Default is 1.
iac	numeric (scalar): Individual auto-correlation for closed cohort sampling on the appropriate scale. Can be used with ICC and CAC instead of tau, eta, rho, gamma, and zeta when the outcome is gaussian. Default is 0 which implies cross-sectional sampling. Values greater than 0 imply closed cohort sampling.
ar	numeric (scalar): Autocorrelation for exponential decay structure between periods within a cluster as described in Kasza et al (2019). Available for cross-sectional sampling only (i.e. zeta=0 and iac=0). See details.
time.lab	character (vector): Labeling for time points when output is display; choice of labeling does not affect results.
silent	logical: if TRUE, hides reminder about order of entries in n when n is not a scalar. Default value is FALSE.
nocheck	logical: if TRUE, various checks on the validity of the arguments are not done. For internal use only.

Details

When simulating from a Gaussian distribution and if eta (and rho) are 0, instead of using tau, eta, rho, gamma and zeta, the icc, cac and iac can be used to define the variability of the random effects. In this model,

$$ICC = \frac{\tau^2 + \gamma^2}{\tau^2 + \gamma^2 + \zeta^2 + \sigma^2}$$

$$CAC = \frac{\tau^2}{\tau^2 + \gamma^2}$$

$$IAC = \frac{\zeta^2}{\zeta^2 + \sigma^2}$$

Choose one parameterization or the other, do not mix parameterizations.

Value

numeric (data frame): Returns the following (individual-level) variables corresponding to the specified SW CRT design:

$response.var	numeric (vector): Response variable based on specified SW CRT design of interest (including family and link function) for each observation in the data frame/set.

`$tx.var`	numeric (vector): Predictor of interest. (Fractional) treatment effect corresponding to 0=control, 1=treatment, and value between 0 and 1 corresponding to treatment arm with fractional treatment effect (for each observation in the data frame/set).
`$timeOnTx.var`	numeric (vector): Predictor of interest when interested in time on treatment lag effect. Total time spent on treatment for each observation in the data frame/set, with 0=control, 1=first time period on treatment, 2=second time period on treatment, etc.
`$time.var`	numeric (vector): Time point id for each observation in the data frame/set.
`$cluster.var`	numeric (vector): Grouping variable. Cluster id for each observation in the data frame/set.

Author(s)

James P Hughes, Navneet R Hakhu, and Emily C Voldal

References

Hussey MA, Hughes JP. Design and analysis of stepped wedge cluster randomized trials. Contemporary Clinical Trials 2007;28:182-191.

Kasza J, Hemming K, Hooper R, Matthews JNS, Forbes AB. Impact of non-uniform correlation on sample size and power in multiple-period cluster randomized trials. Statistical Methods in Medical Research 2019a; 28: 703-716.

Voldal EC, Hakhu NR, Xia, F, Heagerty PJ, Hughes JP. swCRTdesign: An R Package for Stepped Wedge Trial Design and Analysis. Computer Methods and Programs in Biomedicine 2020;196:105514.

Examples

```
library(swCRTdesign)
# Example 1 [ n = scalar; can be vector (for different n for each cluster,
# n=rep(120,22)) or matrix (different n for each cluster at each time point,
# n=matrix(120,22,5)) ]

# generate cross-sectional SW data
design <- swDsn(clusters=c(6,6,6,4))
set.seed(5)
swGenData.nScalar <- swSim( design, family=binomial(link="logit"), n=120,
mu0=log(0.1/0.9), mu1=log(0.9) + log(0.1/0.9), time.effect=0, tau=0.2, eta=0,
rho=0, gamma=0, time.lab=seq(0,12,3))

# summarize SW data by wave
swSummary(response.var, tx.var, time.var, cluster.var, swGenData.nScalar,
type="mean", digits=3)$response.wave

swSummary(response.var, tx.var, time.var, cluster.var, swGenData.nScalar,
type="mean", digits=3)$swDsn
```

swSimPwr *Simulating power for Stepped Wedge Cluster Randomized Trials (SW CRT)*

Description

swSimPwr uses a simulation approach to compute (two-sided) power for the treatment effect(s) from an immediate treatment effect (IT) model or an exposure time indicator (ETI) model (Kenny et al, 2022) for the specified SW CRT design. A cross-sectional or closed cohort sampling scheme can be specified. The outcome of interest can be Gaussian, binomial or Poisson. Individual-level data are generated using swSim for the specified number of clusters per wave, treatment effect (possibly varying according to the exposure time - time after crossing over from control), time effect, family (and link function), number of individuals per cluster per time period, mean in control arm, mean in treatment arm(s), standard deviation (if applicable), standard deviation of random intercepts, standard deviation of random treatment effects, correlation between random intercepts and random treatment effects, standard deviation of random time effects, and, for closed cohorts, standard deviation of individual random effects. For a Gaussian family, standard deviations of random effects may be specified using ICC, CAC and, for closed cohorts, IAC; see swPwr details. These data are then analyzed by either lmer or glmer and the proportion of p-values less than or equal to the specified alpha are computed over the nsim simulations.

Usage

```
swSimPwr(design, family, n, mu0, mu1, time.effect=0, H = NULL,
sigma, tau=0, eta=0, rho=0, gamma=0,zeta=0, icc=0, cac=1, iac=0, ar=1,
alpha = 0.05, nsim = 500, retDATA = FALSE, silent = FALSE, counter=TRUE)
```

Arguments

design	list: A stepped wedge design object, typically from swDsn, that includes at least the following components: swDsn, swDsn.unique.clusters, clusters, n.clusters, total.time, nTxLev, TxLev. Fractional treatment effects specified in swDsn are used in the simulation.
family	character: Used in typical way. Only Gaussian, Binomial, and Poisson families accepted. Also, only identity, logit, and log links accepted. Logit link is only available for Binomial family, and log link is only available for Binomial and Poisson. Currently, 'Binomial' implies Bernoulli. Default links are identity for Gaussian, logit for Binomial, log for Posson. ***NOTE: It is the users responsibility to make sure specified parameters (mu0, mu1, time.effect, tau, eta, rho, gamma, zeta) are ALL on the SAME scale as the specified link function; see example.***
n	integer (scalar, vector, or matrix): Number of observations: (scalar) for all clusters and all time points; (vector) for each cluster at all time points; and (matrix) for each cluster at each time point, where rows correspond to clusters and columns correspond to time. n can also be used to specify a design with transition periods were no data is collected (a transition period is a period, typically,

immediately after treatment is introduced were no data are collected). Simply define n as a matrix with a sample size of 0 during every transition period. This is equivalent to putting an NA in design$swDsn.

`mu0` numeric (scalar): Mean outcome in the control group on the appropriate scale.

`mu1` numeric (scalar or vector): Mean outcome in the treatment group on the appropriate scale. If design$nTxLev > 1 (number of treatment levels greater than 1) then mu1 should have length equal to design$nTxLev; if design$nTxLev = 1 then mu1 should be either a scalar (constant treatment effect) or a vector with length equal to the maximum number of exposure times (time-varying treatment effect). Do not specify time-varying or multiple treatment effects if the design includes fractional treatment effects.

`time.effect` integer (scalar or vector): Time effect at each time point on the appropriate scale (added to mean at each time). Default is 0.

`H` numeric (vector): If NULL, then swPwr assumes an immediate, constant treatment effect (IT) model. If not NULL, then an exposure time indicator (ETI) model is assumed and H should be a vector as long as the longest exposure time (typically, number of time periods minus one). H specifies the desired linear combination of exposure time treatment effects. For example, in a stepped wedge trial with 5 time periods and four exposure times, H = rep(.25,4) gives the average treatment effect over the four exposure times; H = c(0,0,.5,.5) ignores the first two periods after the intervention is introduced and averages the remaining periods. Typically, the sum of H is 1.0; if not, it is renormalized to sum to 1.0. H can only be specified when there is a single intervention level (i.e. design$swDsn includes only NA,0,1; see cautions in help for `swDsn` about including NA periods when H is used).

`sigma` numeric (scalar): Pooled treatment and control arm standard deviation on the appropriate scale. Ignored if family != Gaussian.

`tau` numeric (scalar): Standard deviation of random intercepts on the appropriate scale. Default is 0.

`eta` numeric (scalar): Standard deviation of random treatment effects on the appropriate scale. Default is 0.

`rho` numeric (scalar): Correlation between random intercepts and random treatment effects on the appropriate scale. Default is 0.

`gamma` numeric (scalar): Standard deviation of random time effects on the appropriate scale. Default is 0.

`zeta` numeric (scalar): Standard deviation of individual effects for closed cohort sampling on the appropriate scale. Default is 0 which implies cross-sectional sampling. Values greater than 0 imply closed cohort sampling.

`icc` numeric (scalar): Within-period intra-cluster correlation on the appropriate scale. Can be used with CAC instead of tau, eta, rho, and gamma when the outcome is Gaussian. Default is 0.

`cac` numeric (scalar): Cluster auto-correlation on the appropriate scale. Can be used with ICC instead of tau, eta, rho, and gamma when the outcome is Gaussian. Default is 1.

iac	numeric (scalar): Individual auto-correlation for closed cohort sampling on the appropriate scale. Can be used with ICC and CAC instead of tau, eta, rho, gamma, and zeta when the outcome is gaussian. Default is 0 which implies cross-sectional sampling. Values greater than 0 imply closed cohort sampling.
ar	numeric (scalar): Autocorrelation for exponential decay structure between periods within a cluster as described in Kasza et al (2019a;2019b). Available for cross-sectional sampling only (i.e. zeta=0 and iac=0). Default is ar=1 (no exponential decay). See details.
alpha	numeric (scalar): Desired alpha level for a two-sided test.
nsim	numeric (scalar): Number of simulations.
retDATA	logical: if FALSE, only power is returned; if TRUE, then estimates, standard errors, p-values and convergence information from all simulations are also returned.
silent	logical: if TRUE, hides most warning messages. Default value is FALSE.
counter	logical: if TRUE, show a counter for the simulations. Default value is TRUE

Details

swSimPwr uses a simulation approach to compute the two-sided statistical power for testing the hypotheses $Ho : \mu_1 - \mu_0 = 0$ if H is NULL or $Ho : \sum H*(\mu_1 - \mu_0) = 0$ if H is non-NULL. If ar=1 then swSim is called to generate nsim simulations and then lmer or glmer is used to analyze each simulated dataset. If ar!=1 then swSim2 is called to generate nsim simulations and then glmmTMB is used to analyze each simulated dataset. The simulated power is equal to the proportion of p-values from lmer/glmer that are less than or equal to alpha. For small sample sizes it is recommended that the user run a simulation under the null to check the accuracy of the simulated type I error rate. Optionally, a counter is printed as each simulation is completed.

Value

numeric (vector): if retDATA = FALSE, swSimPwr returns the power for the two-sided design$nTxLev hypothesis test(s) treatment effect(s) against a null of 0.0.

numeric (list): if retDATA = TRUE, swSimPwr returns the following objects of a list.

power	Power for the two-sided design$nTxLev hypothesis test(s) of treatment effect(s) against a null of 0.0.
nsim	Number of simulation power is based on
est	matrix: nsim x design$nTxLev matrix of estimated treatment effects. If H is specified then this is the linear combination of the exposure time estimates.
se	matrix: nsim x design$nTxLev matrix of standard errros of the estimated treatment effects. If H is specified then this is the standard error of the linear combination of the exposure time estimates.
sdcor	matrix: matrix with nsim rows and number of columns equal to the number of variance components in the model (expressed on the random effects scale) giving the estimated variance components (expressed as standard deviations) from each fit. Note that eta is expressed as a correlation.

df	matrix: nsim x design$nTxLev matrix of degrees of freedom for testing the hypotheses on the treatment effect(s). Only returned if lmer is called for estimation; otherwise NULL.
pval	matrix: nsim x design$nTxLev matrix of p-values for testing the hypotheses on the treatment effect(s).
code	vector: vector of length nsim giving either 0 (convergence) or the value of rslt@optinfo$conv$lme4$code for each simulation
msg	vector: vector of length nsim giving all messages from rslt@optinfo$conv$lme4$messages for each simulation

References

Kenny A, Voldal E, Xia F, Heagerty PJ, Hughes JP. Analysis of stepped wedge cluster randomized trials in the presence of a time-varying treatment effect. Statistics in Medicine, in press, 2022.

Examples

```
logit = function(x){log(x/(1 - x))}
stdy1 <- swDsn(c(6,6,6,6),swBlk=matrix(c(0,1,1,1,1,0,0,1,1,1,0,0,0,1,1,0,0,0,0,1),4,5,byrow=TRUE))
# analytic power by swPwr
swPwr(stdy1, distn="binomial",n=120, mu0=0.45, mu1=0.5,
      tau=0.1, alpha=0.05, retDATA=FALSE)
#
# equivalent analysis on logit scale with transformation of variables
swGlmPwr(stdy1,distn="binomial",n=120,fixed.intercept=logit(0.45),
         fixed.treatment.effect = logit(0.5)-logit(0.45), fixed.time.effect=0.0,
         tau=.1/(.475*.525))
#
# Same problem by simulation using three different approaches
#
#swSimPwr(stdy1,family=binomial(link="logit"),n=120,
#         mu0=logit(.45),mu1=logit(.5),time.effect=0.0,
#         tau=.1/(.475*.525),nsim=500)
#
#swSimPwr(stdy1,family="gaussian",n=120,
#         mu0=.45,mu1=.5,time.effect=0.0,
#         sigma=sqrt(.475*.525),tau=.1,nsim=500)
#
#swSimPwr(stdy1,family=binomial(link="identity"),n=120,
#         mu0=.45,mu1=0.5,time.effect=0.0,
#         tau=.1,nsim=500)
```

---

swSiz *Sample Size of Stepped Wedge Cluster Randomized Trial (SW CRT) for LMM*

Description

swSiz is a wrapper function for swPwr. For a given design, number of clusters, power and fixed and random effect parameters, swSiz computes the number of individuals per cluster needed to achieve the desired power.

Usage

```
swSiz(power, alpha=0.05, ...)
```

Arguments

power	numeric (scalar) Desired power. Must be between α (default = 0.05) and 1.
alpha	numeric (scalar): Two-sided statistical significance level.
...	Arguments to be passed to swPwr. See documentation for swPwr.

Details

swSiz calls swPwr iteratively to find the number of participants per cluster-period needed to achieve power power. Note that all warnings and messages normally generated by swPwr are turned off. Also, the argument retDATA is set to FALSE. It is recommended that after using swSiz, the user call swPwr with the recommended sample size to view any warning messages and/or to get the data normally returned when retDATA is TRUE.

An error in swPwr causes swSiz to terminate and the error message from swPwr is printed.

Value

numeric (list): swSiz returns a list of the following:

n	scalar: number of individuals per cluster-period required to achieve the desired power.
power	scalar: exact power, as computed by swPwr, obtained with the sample size n.

Author(s)

Avi Kenny, James P Hughes

Examples

```
swSiz(power = 0.9, design = swDsn(clusters=c(4,4,4,4)),
  distn = "gaussian", mu0 = 0, mu1 = 0.1, sigma = 0.5, icc = 0.1)
```

swSummary *Summary of Response/Outcome for Stepped Wedge Cluster Randomized Trial (SW CRT)*

Description

swSummary returns the mean, sum, and number of non-missing values for the response/outcome variable of interest for each cluster at each time point from a SW CRT.

Usage

```
swSummary(response.var, tx.var, time.var, cluster.var, data,
type="mean", imputeNA=FALSE, digits=16, fcn.Call=FALSE)
```

Arguments

response.var	numeric(vector): Response (Outcome) variable.
tx.var	numeric(vector): Treatment (Predictor of Interest) variable. Typically, 0=placebo, 1=intervention, and values greater than 1 correspond to other treatment levels.
time.var	integer(vector): Time (points) variable, corresponding to the time points when data were collected during the SW CRT.
cluster.var	integer(vector) or character(vector): Cluster (identification) variable, corresponding to the cluster where an individual is from.
data	An optional data frame containing (at least) the response, treatment (tx), time, and cluster variables.
type	character (scalar): Specify which summary measure is of interest from "mean", "sum", and "n". (Note: The default returns "mean" as the summary measure of response. Note that all summary measures may not be scientifically relevant in every situation.)
imputeNA	logical (scalar): If TRUE, then swSummary attempts to impute the intervention status in $swDsn and $swDsn.unique.clusters for cluster-periods with 0 non-missing observations (see Details). If FALSE, cluster-periods with 0 non-missing observations are assigned NA in $swDsn and $swDsn.unique.clusters. Note that changing imputeNA from FALSE to TRUE may also change the inferred number of unique waves ($n.waves and $swDsn.waves). Default is FALSE.
digits	integer (scalar): Number of places right of the decimal. The default value is 16.
fcn.Call	logical: Only TRUE when calling swSummary from within swPlot. The default is FALSE.

Details

If imputeNA is set to TRUE then any NA in the design matrix is set to 1 (intervention) if the NA follows a 1 in the same cluster; is set to 0 (control) if the NA precedes a 0 in the same cluster; and is otherwise left as NA.

swSummary returns a list containing a matrix of dimension length(unique(data$cluster)) by length(unique(data$time)) that summarizes data$response for specified type. Either the mean, sum, or the number of non-missing data$response values may be requested using type. dimnames of the matrix correspond to the unique values of cluster and time. Note that the stepping pattern in the data may be obtained by specifying the treatment variable name as the response and type = "mean".

Value

numeric (list): swSummary returns a list containing the following

type	One of user-specified options "mean", "sum", or "n".
swDsn	The SW design.
swDsn.unique.clusters	The unique clusters (i.e., rows) SW design.
swDsn.waves	Wave for each cluster.
n.waves	Number of waves. Note that this is usually, but not necessarily, the same as data$n.waves. See example 2.
clusters	Clusters per wave.
n.clusters	Total number of clusters.
time.at.each.wave	Time at each wave.
total.time	Total time points.
response.cluster	numeric (matrix): Response variable summarized according to type for all clusters, with dimension length(data$cluster) by length(unique(data$time)).
response.wave	numeric (matrix): Response variable summarized according to type, for all waves (all clusters of a particular wave are combined), with dimension length(unique(data$cluster)) by length(unique(data$time)).

Author(s)

James P Hughes and Navneet R Hakhu

References

Hussey MA, Hughes JP. Design and analysis of stepped wedge cluster randomized trials. Contemporary Clinical Trials 2007;28:182-191.

Examples

```
library(swCRTdesign)
# Example 1
# Generate binary response with 5 clusters, 4 time points
data.Ex1 <- swSim(swDsn(c(2,2,1)),family=binomial(link="identity"), n=120,
                    mu0=0.25, mu1=0.30,time.effect=0, tau=0.05, gamma=.01)

## Example 1 (type="mean", by cluster and by wave)
```

```
swSummary.Ex1.mean <- swSummary(response.var, tx.var, time.var, cluster.var,
data=data.Ex1, type="mean", digits=3)

swSummary.Ex1.mean$response.cluster

swSummary.Ex1.mean$response.wave

# Example 1 (type="sum", by cluster and by wave)
swSummary.Ex1.sum <- swSummary(response.var, tx.var, time.var, cluster.var,
data=data.Ex1, type="sum")

swSummary.Ex1.sum$response.cluster

swSummary.Ex1.sum$response.wave

## Example 1 (type="n", by cluster and by wave)
swSummary.Ex1.n <- swSummary(response.var, tx.var, time.var, cluster.var,
data=data.Ex1, type="n")

swSummary.Ex1.n$response.cluster

swSummary.Ex1.n$response.wave

## Example 2
design2 <- swDsn(clusters=c(6,6,6,6))
# design2$n.waves says 4 waves
nmat2 = rbind(matrix(rep(c(120,120,120,120,120),6),6,5,byrow=TRUE),
              matrix(rep(c(120,120,120,120,120),3),3,5,byrow=TRUE),
              matrix(rep(c(0,120,120,120,120),3),3,5,byrow=TRUE),
              matrix(rep(c(0,0,120,120,120),6),6,5,byrow=TRUE),
              matrix(rep(c(0,0,0,120,120),6),6,5,byrow=TRUE))
swGenData2 <- swSim( design2,family=binomial(link="logit"), n=nmat2,
                     mu0=log(0.1/0.9), mu1=log(0.9) + log(0.1/0.9),
                     time.effect=0, tau=0.2, time.lab=seq(0,12,3))
swSummary(response.var, tx.var, time.var, cluster.var, swGenData2,
          type="mean", digits=3)
# This summary create 5 waves
#
# The following is equivalent to above
design2b <- swDsn(c(6,3,3,6,6),
                  swBlk=matrix(c(0,1,1,1,1,
                                 0,0,1,1,1,
                                 NA,0,1,1,1,
                                 NA,NA,0,1,1,
                                 NA,NA,NA,0,1),5,5,byrow=TRUE))
# design2b$n.waves says 5 waves
swGenData2b <- swSim(design2b,family=binomial(link="logit"), n=120,
                     mu0=log(0.1/0.9), mu1=log(0.9) + log(0.1/0.9),
                     time.effect=0, tau=0.2, time.lab=seq(0,12,3))
swSummary(response.var, tx.var, time.var, cluster.var, swGenData2b,
          type="mean", digits=3)
# This summary create 5 waves
```

Index

∗ **block diagonal matrix**
  blkDiag, 3
∗ **block diagonal**
  swCRTdesign-package, 2
∗ **closed cohort sampling**
  swCRTdesign-package, 2
∗ **cluster randomized trial**
  swCRTdesign-package, 2
  swDsn, 4
  swGlmPwr, 8
  swGlmSiz, 14
  swPlot, 15
  swPwr, 17
  swSim, 24
  swSim2, 27
  swSiz, 34
  swSummary, 36
∗ **design**
  swCRTdesign-package, 2
  swDsn, 4
  swGlmPwr, 8
  swSim, 24
  swSim2, 27
∗ **discrete outcomes**
  swGlmPwr, 8
∗ **plot**
  swCRTdesign-package, 2
  swPlot, 15
∗ **power**
  swCRTdesign-package, 2
  swGlmPwr, 8
  swGlmSiz, 14
  swPwr, 17
  swSiz, 34
∗ **repeated cross-sectional sampling**
  swCRTdesign-package, 2
∗ **simulated data set**
  swCRTdesign-package, 2
∗ **stepped wedge**
  swCRTdesign-package, 2
  swDsn, 4
  swGlmPwr, 8
  swGlmSiz, 14
  swPlot, 15
  swPwr, 17
  swSim, 24
  swSim2, 27
  swSiz, 34
  swSummary, 36
∗ **summary**
  swCRTdesign-package, 2
  swSummary, 36

blkDiag, 3

Release Notes, 4

swCRTdesign *(*swCRTdesign-package*)*, 2
swCRTdesign-package, 2
swDsn, 4
swGlmPwr, 8
swGlmSiz, 14
swPlot, 15
swPwr, 17, *24*, *27*, *31*
swSim, 24
swSim2, 27
swSimPwr, 31
swSiz, 34
swSummary, 36