Package ‘spass’

October 1, 2016

Title Study Planning and Adaptation of Sample Size

Version 1.0

Date 2016-09-12

Author T. Asendorf, R. Gera, S. Islam, M. Harden and M. Placzek

Maintainer Marius Placzek <marius.placzek@med.uni-goettingen.de>

Description Sample size estimation and blinded sample size reestimation in Adaptive Study Design.

Imports stats, utils

Depends mvtnorm,multcomp,Rcpp,MASS,geepack

License GPL

LazyData true

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-10-01 01:00:37

R topics documented:

bssr.1subgroup *Blinded Sample Size Recalculation for a One Subgroup Design*

Description

Given data from an Internal Pilot Study (IPS), `bssr.1subgroup` reestimates the nuisance parameters, i.e. variances and prevalence, and recalculates the required sample size for proving a desired alternative when testing for an effect in the full or subpopulation. See 'Details' for more information.

Usage

```
bssr.1subgroup(data, alpha, beta, delta, eps = 0.001,
  approx = c("conservative.t", "liberal.t", "normal"), df = c("n", "n1"),
  adjust = c("YES", "NO"), k = 1, nmax = 1000)
```

Arguments

`data`	data matrix with data from ongoing trial: see 'Details'.
`alpha`	level (type I error) to which the hypothesis is tested.
`beta`	type II error (power=1-beta) to which an alternative should be proven.
`delta`	vector of treatment effects to be proven, c(outside subgroup, inside subgroup).
`eps`	precision parameter concerning the power calculation in the iterative sample size search algorithm.
`approx`	approximation method: Use a conservative multivariate t distribution ("conservative.t"), a liberal multivariate t distribution ("liberal.t") or a multivariate normal distribution ("normal") to approximate the joint distribution of the standardized test statistics.
`df`	in case of a multivariate t distribution approximation, recalculate sample size with degrees of freedom depending on the size of the IPS (df=n1) or depending on the final sample size (df=n).
`adjust`	adjust blinded estimators for assumed treatment effect ("YES","No").
`k`	sample size allocation factor between groups: see 'Details'.
`nmax`	maximum total sample size.

Details

This function performs blinded nuisance parameter reestimation in a design with a subgroup within a full population where we want to test for treatment effects between a control and a treatment group. Then the required sample size for the control and treatment group to prove an existing alternative `delta` with a specified power 1-`beta` when testing the global null hypothesis $H_0 : \Delta_F = \Delta_S = 0$ to level `alpha` is calculated.

The data matrix `data` should have three columns: The first column has to be a binary variable (0=treatment group, 1=control group). The second column should also contain a binary variable giving the full population/subgroup differentiation (0=full population, 1=subpopulation). The last column contains the observations.

For sample sizes n_C and n_T of the control and treatment group, respectively, the argument `k` is the sample size allocation factor, i.e. $k = n_T / n_C$.

The parameter `df` provides a difference to the standard sample size calculation procedure implemented in `n.1subgroup`. When applying a multivariate t distribution approximation to approximate the joint distribution of the standardized test statistics it gives the opportunity to use degrees of freedom depending on the number of subjects in the IPS instead of degrees of freedom depending on the projected final sample size. Note that this leads to better performance when dealing with extremely small subgroup sample sizes but significantly increases the calculated final sample size.

Value

`bssr.1subgroup` returns a list containing the recalculated required sample size within the control group and treatment group along with all relevant parameters. Use `summary.bssrest` for a structured overview.

Source

`bssr.1subgroup` uses code contributed by Marius Placzek.

See Also

`n.1subgroup` for sample size calculation prior to the trial.

Examples

```
#Given data from the Internal Pilot Study, reestimate the nuisance parameters and
#recalculate the required sample size to correctly reject with
#80% probability when testing the global Nullhypothesis H_0: Delta_F=Delta_S = 0
#assuming the true effect Delta_S=1 is in the subgroup (no effect outside of the subgroup).

random<-r.1subgroup(n=50, delta=c(0,1), sigma=c(1,1.2), tau=0.4, fix.tau="YES", k=2)
reestimate<-bssr.1subgroup(data=random,alpha=0.05,beta=0.1,delta=c(0,1),eps=0.001,
approx="conservative.t",df="n1",k=2,adjust="NO")
summary(reestimate)
```

`bssr.gee.1subgroup` *Blinded Sample Size Recalculation for longitudinal data in a One Subgroup Design*

Description

Given data from an Internal Pilot Study (IPS), `bssr.GEE.1subgroup` given the reestimated nuisance parameteres are calculated. `bssr.gee.1subgroup` is a wraper for `n.gee.1subgroup` because the reestimation of the variances can be highly dependable on the user and should be done seperatly. see "detail" for more information on that.

Usage

```
bssr.gee.1subgroup(alpha, tail = "both", beta = NULL, delta, estsigma,
  tau = 0.5, k = 1)
```

Arguments

`alpha` level (type I error) to which the hypothesis is tested.

`tail` which type of test is used, e.g. which quartile und H0 is calculated

`beta` type II error (power=1-beta) to which an alternative should be proven.

`delta` vector of regression coefficients values which shall be proven, c(allcomers, only subpopulation).

`estsigma` reestimated vector of assymptotic standard deviations.

`tau` ration between F/S and S

`k` sample size allocation factor between groups: see 'Details'.

Details

This function provides a simple warper for `n.gee.1subgroup` where instead of initial assumptions blind estimated nuisance parameter inserted. For information see `n.gee.1subgroup`. alternative `delta` with a specified power 1-`beta` when testing the global null hypothesis $H_0 : \beta_3^F = \beta_3^S = 0$ to level `alpha` is calculated.

The data matrix `data` should have as many columns as observed timepoints: first column first observed timepoint. As of now the timepoints must be equispaced to calculate the correct intra-subject covariance-matrix. Entries can be `NA`. See `r.gee.1subgroup.r` for more information.

For sample sizes n_C and n_T of the control and treatment group, respectively, the argument `k` is the sample size allocation factor, i.e. $k = n_T/n_C$.

Value

`bssr.gee.1subgroup` returns a list containing the recalculated required sample size within the control group and treatment group along with all relevant parameters. Use `summary.bssrest` for a structured overview.

Source

bssr.gee.1subgroup uses code contributed by Roland Gerard Gera.

See Also

n.gee.1subgroup for sample size calculation prior to the trial, r.gee.1subgroup how list data should look like and estimcov how the reestimation of nuisance parameters works. See *sim.gee* for an enxample for an initial sample size estimation and reestimation to see the functions working in junction.

Examples

```
estimate<-bssr.gee.1subgroup(alpha=0.05,beta=0.2,delta=c(0.1,0.1),estsigma=c(0.8,0.4),tau=0.4, k=1)
summary(estimate)
```

bssr.nb.inar1 *Blinded Sample Size Reestimation for Longitudinal Count Data using the NB-INAR(1) Model*

Description

bssr.nb.inar1 fits blinded observations and recalculates the sample size required for proving a desired alternative when testing for a rate ratio between two groups unequal to one. See 'Details' for more information.

Usage

```
bssr.nb.inar1(alpha, power, delta, x, n, k)
```

Arguments

alpha level (type I error) to which the hypothesis is tested.

power power (1 - type II error) to which an alternative should be proven.

delta the rate ratio which is to be proven.

x a matrix or data frame containing count data which is to be fitted. Columns correspond to time points, rows to observations.

n a vector giving the sample size within the control group and the treatment group, respecitvely.

k planned sample size allocation factor between groups: see 'Details'.

Details

When testing for differences between rates μ_C and μ_T of two groups, a control and a treatment group respectively, we usually test for the ratio between the two rates, i.e. $\mu_T/\mu_C = 1$. The ratio of the two rates is refered to as δ, i.e. $\delta = \mu_T/\mu_C$.

bssr.nb.inar1 gives back the required sample size for the control and treatment group required to prove an existing alternative theta with a specified power power when testing the null hypothesis $H_0 : \mu_T/\mu_C \geq 1$ to level alpha. Nuisance parameters are estimated through the blinded observations x, thus not further required.

for sample sizes n_C and n_T of the control and treatment group, respectively, the argument k is the desired sample size allocation factor at the end of the study, i.e. $k = n_T/n_C$.

Value

rnbinom.inar1 returns the required sample size within the control group and treatment group.

Source

rnbinom.inar1 uses code contributed by Thomas Asendorf.

See Also

rnbinom.inar1 for information on the NB-INAR(1) model, n.nb.inar1 for calculating initial sample size required when performing inference, fit.nb.inar1 for calculating initial parameters required when performing sample size estimation

Examples

```
#Calculate required sample size to find significant difference with
#80% probability when testing the Nullhypothesis H_0: mu_T/mu_C >= 1
#assuming the true effect delta is 0.8 and rate, size and correlation
#parameter in the control group are 2, 1 and 0.5, respectively.

estimate<-n.nb.inar1(alpha=0.025, power=0.8, delta=0.8, muC=2, size=1, rho=0.5, tp=7, k=1)

#Simulate data
placebo<-rnbinom.inar1(n=50, size=1, mu=2, rho=0.5, tp=7)
treatment<-rnbinom.inar1(n=50, size=1, mu=1.6, rho=0.5, tp=7)

#Blinded sample size reestimation
blinded.data<-rbind(placebo, treatment)[sample(1:100),]
estimate<-bssr.nb.inar1(alpha=0.025, power=0.8, delta=0.8, x=blinded.data, n=c(50,50), k=1)
summary(estimate)
```

estimcov *Estimation of variance, intra-subject-correlation and dropout*

Description

estimcov estimates variance, intra-subject-correlation and dropout given empirical data.

Usage

```
estimcov(data, Time, Startvalues = c(3, 0.5, 1), stepwidth = c(0.001, 0.001,
  0.001), maxiter = 10000, lower = c(1e-04, 1e-04, 1e-04), upper = c(Inf,
  5, 3))
```

Arguments

data	list of gathered data. The list must be consistent with the gernerated data of r.gee.1subgroup
Time	list with observed time points: see 'Details'
Startvalues	startvalues for the paramteres var,rho and theta
stepwidth	vector of stepwidths which the optimisation-function should use
maxiter	value setting maximal amount of iterations for the optimisation algorithm
lower	lower bound for var,rho and theta
upper	upper bound for var,rho and theta

Details

Function estimcov fits a covariance-matrix with parameters var,rho and theta (see gen_cov_cor for matrix generation) to an empirical covarince-matrix provided by data.

Value

estimcov returns a list with two vectors. The first entry consists of a vector with estimations for c(var,rho,theta) while the second entry contains a vector, describing the empirical dropout-chance per timepoint.

Source

estimcov uses code contributed Roland Gerard Gera.

See Also

r.gee.1subgroup for information on the generated longitudinal data and n.gee.1subgroup for the calculation of initial sample sizes for longitudinal GEE-models and bssr.gee.1subgroup for blinded sample size reestimation within a trial. See gen_cov_cor for more information on the generation of covariance matrices.

Examples

```
#Generate data from longitudinal-model
set.seed(2015)
dataset<-r.gee.1subgroup(n=300, reg=list(c(0,0,0,0.1),c(0,0,0,0.1)), sigma=c(3,2.5), tau=0.5,
rho=0.25, theta=1, k=1.5, Time=c(0:5), OD=0.2)

estimate<-estimcov(data=dataset,Time=c(0:5))
estimate
```

---

fit.nb.inar1 *Fitting Longitudinal Data with Negative Binomial Marginal Distribution and Autoregressive Correlation Structure of Order One: NB-INAR(1)*

Description

`fit.nb.inar1` fits data using the maximum likelihood of a reparametrized NB-INAR(1) model.

Usage

```
fit.nb.inar1(x, lower = rep(10, 3)^-5, upper = c(10^5, 10^5, 1 - 10^-5),
  method = "L-BFGS-B", start)
```

Arguments

`x`	a matrix or data frame containing count data which is to be fitted. Columns correspond to time points, rows to observations.
`lower`	vector of lower bounds for estimated parameters `mu`, `size` and `rho`, respectively.
`upper`	vector of upper bounds for estimated parameters `mu`, `size` and `rho`, respectively.
`method`	algorithm used for minimization of the likelihood, see `optim` for details.
`start`	vector of starting values for estimated parameters `mu`, `size` and `rho`, respectively, used for optimization.

Details

the function `fit.nb.inar1` fits a reparametrization of the NB-INAR(1) model as found in McKenzie (1986). The reparametrized model assumes equal means and dispersion parameter between time points with an autoregressive correlation structure. The function is especially useful for estimating parameters for an initial sample size calculation using `n.nb.inar1`. The fitting function allows for incomplete follow up, but not for intermittent missingness.

Value

`fit.nb.inar1` return estimates of the mean `mu`, dispersion parameter `size` and correlation coefficient `rho`.

Source

fit.nb.inar1 uses code contributed by Thomas Asendorf.

References

McKenzie Ed (1986), Autoregressive Moving-Average Processes with Negative-Binomial and Geometric Marginal Distributions. *Advances in Applied Probability* Vol. 18, No. 3, pp. 679-705.

See Also

rnbinom.inar1 for information on the NB-INAR(1) model, n.nb.inar1 for calculating initial sample size required when performing inference, bssr.nb.inar1 for blinded sample size reestimation within a running trial, optim for more information on the used minimization algorithms.

Examples

```
#Generate data from the NB-INAR(1) model
set.seed(8)
random<-rnbinom.inar1(n=1000, size=1.5, mu=2, rho=0.6, tp=7)

estimate<-fit.nb.inar1(random)
estimate
```

gen_cov_cor *Generation of covariance- or correlation-matrices*

Description

Generate covariance- or correlation-matrices given the parameters var, rho, theta for the covariance structure, Time for the observed timepoints and cov=TRUE if a covariance or cov=FALSE if a correlation-matrix is to be generated.

Usage

```
gen_cov_cor(var = 1, rho, theta, Time, cov = TRUE)
```

Arguments

var variance at each timepoint

rho correlation between two adjacent timepoints 1 timeunit appart

theta variable specifying the type of the correlation structure: see 'Details'

Time list with time measures which are used to generate the covariance- or correlation-structure: see 'Details'

cov TRUE/FALSE statement which determines if a covariance- or a correlation-matrix is generated.

Details

The function gen_cov_cor is used to generate either a covariance- or a correlation-matrix. Given vector Time and parameters var, rho and theta the covariance or correlation between two timepoints is described: The way the correlation between two timepoints is described by

$cov(Time[i], Time[j]) = var * (rho^{(}abs(Time[i] - Time[j])^{t}heta))$

for covariance and

$corr(Time[i], Time[j]) = rho^{(}abs(Time[i] - Time[j])^{t}heta)$

for correlation. [The above sentence would be better is we write the following sentence:] [[The following two equations are used to calculate the covariance and the correlation between two timepoints, respectively: cov(Time[i],Time[j])=var*(rho^(abs(Time[i]-Time[j])^theta)) corr(Time[i],Time[j])=rho^(abs(Time[i]-Time[j])^theta)]]

Which of the two formulas is used depends on cov

Value

gen_cov_cor returns the covariance- or correlation-matrix, depending if cov was TRUE or FALSE.

Source

gen_cov_cor uses code contributed by Roland Gerard Gera

@seealso r.gee.1subgroup for information on the generated longitudinal data and n.gee.1subgroup for the calculation of initial sample sizes for longitudinal GEE-models and bssr.gee.1subgroup for blinded sample size reestimation within a trial See estimcov for more information on the used minimization algorithms.

Examples

```
#Generate a covariance-matrix with measurements at Baseline and at timeunit 1,1.5,2
#and 5 (hours,day,months,years, etc.)

covar<-gen_cov_cor(var=3,rho=0.25,theta=1,Time=c(0,1,1.5,2,5),cov=TRUE)
covar

#Generate a correlation-matrix

corr<-gen_cov_cor(rho=0.25,theta=1,Time=c(0,1,1.5,2,5),cov=FALSE)
corr
```

n.1subgroup *Sample Size Calculation for a One Subgroup Design*

Description

n.1subgroup calculates the required sample size for proving a desired alternative when testing for an effect in the full or subpopulation. See 'Details' for more information.

Usage

```
n.1subgroup(alpha, beta, delta, sigma, tau, eps = 0.001,
  approx = c("conservative.t", "liberal.t", "normal"), k = 1, nmax = 1000,
  nmin = 0)
```

Arguments

`alpha` level (type I error) to which the hypothesis is tested.

`beta` type II error (power=1-beta) to which an alternative should be proven.

`delta` vector of treatment effects to be proven, c(outside subgroup, inside subgroup).

`sigma` vector of standard deviations, c(outside subgroup, inside subgroup).

`tau` subgroup prevalence.

`eps` precision parameter concerning the power calculation in the iterative sample size search algorithm.

`approx` approximation method: Use a conservative multivariate t distribution ("conservative.t"), a liberal multivariate t distribution ("liberal.t") or a multivariate normal distribution ("normal") to approximate the joint distribution of the standardized teststatistics.

`k` sample size allocation factor between groups: see 'Details'.

`nmax` maximum total sample size.

`nmin` minimum total sample size.

Details

This function performs sample size estimation in a design with a subgroup within a full population where we want to test for treatment effects between a control and a treatment group. Since patients from the subgroup might potentially benefit from the treatment more than patients not included in that subgroup, one might prefer testing hypothesis cercerning the full population and the subpopulation at the same time. Here standardized test statistics are their joined distributions are used to calculate the required sample size for the control and treatment group to prove an existing alternative `delta` with a specified power 1-`beta` when testing the global null hypothesis $H_0 : \Delta_F = \Delta_S = 0$ to level `alpha`.

For sample sizes n_C and n_T of the control and treatment group, respectively, the argument `k` is the sample size allocation factor, i.e. $k = n_T/n_C$.

Value

`n.1subgroup` returns the required sample size within the control group and treatment group.

Source

`n.1subgroup` uses code contributed by Marius Placzek.

See Also

#' `bssr.1subgroup` for blinded sample size reestimation within a running trial.

Examples

```
#Calculate required sample size to correctly reject with
#80% probability when testing the global Nullhypothesis H_0: Delta_F=Delta_S = 0
#assuming the true effect Delta_S=1 is in the subgroup (no effect outside of the subgroup)
#with subgroup prevalence tau=0.4.
#The variances in and outside of the subgroup are unequal, sigma=c(1,1.2).

estimate<-n.1subgroup(alpha=0.025,beta=0.1,delta=c(0,1),sigma=c(1,1.2),tau=0.4,eps=0.0001,
approx="conservative.t",k=2)
summary(estimate)
```

---

n.gee.1subgroup *Sample Size estimation for longitudinal GEE Models when testing 1 coefficient*

Description

n.gee.1subgroup calculates the required sample size for proving a desired alternative when testing regression coefficients in the full or subpopulation. See 'Details' for more information.

Usage

```
n.gee.1subgroup(alpha, tail = "both", beta = NULL, delta, sigma,
  tau = 0.5, k = 1, npow = NULL, nmax = Inf)
```

Arguments

alpha level (type I error) to which the hypothesis is tested.

tail which type of test is used, e.g. which quartile und H0 is calculated

beta type II error (power=1-beta) to which an alternative should be proven.

delta vector of regression coefficients values which shall be proven, c(allcomers, subpopulation).

sigma vector of assymptotic standard diviation of regressors, c(full population, subpopulation).see 'Details'

tau subgroup prevalence.

k sample size allocation factor between control and treatment: see 'Details'.

npow calculates power of a test if npow is a sample size

nmax maximum total sample size.

Details

This function performs sample size estimation in a design with a subgroup nested within a full population. To calculate the required sample size when testing only one regressor (e.g. effect of treatment*time) one needs to input the expected value of the regressor under alternative, delta, and the expected asymptotic variance of that regressor, sigma. The power for the global null hypothesis is given by 1-beta and alpha specifies the false positve level for rejecting $H_0 : \Delta_F = \Delta_S = 0$ to level alpha, where in our context $\Delta_F and \Delta_S$ normaly represent regressioncoefficents and σ^2 their variance.

For sample sizes n_C and n_T of the control and treatment group, respectively, the argument k is the sample size allocation factor, i.e. $k = n_T/n_C$.

Value

n.gee.1subgroup returns the required sample size within the control group and treatment group.

Source

n.gee.1subgroup uses code contributed by Roland Gerard Gera.

See Also

bssr.1subgroup for blinded sample size reestimation within a running trial and sandwich for estimating asymptotic covarianc mtrices in GEE models.

Examples

```
#Calculate required sample size to correctly reject with
#80% probability when testing the global Nullhypothesis H_0: Delta_F=Delta_S = 0
#assuming the coefficient in and outside of the subgroup is Delta=c(0.1,0,1) with a
#subgroup-prevalence of tau=0.4.
#The assymptotic variances in and outside of the subgroup are unequal, sigma=c(0.8,0.4).

estimate<-n.gee.1subgroup(alpha=0.05,beta=0.2,delta=c(0.1,0.1),sigma=c(0.8,0.4),tau=0.4, k=1)
summary(estimate)

#Now we want to estimate the power our study would have,
#if we know the effect in and outside the subgroup, as
#well as asymptotic variance of the regressors. Here we
#estimate that only 300 Patiens total can be recruited.
#All other parameters are the same as those above.


n.gee.1subgroup(alpha=0.05,delta=c(0.1,0.1),sigma=c(0.8,0.4),tau=0.4, k=1, npow=300)
```

`n.nb.inar1` *Sample Size Calculation for Comparing Two Groups when observing Longitudinal Count Data with marginal Negative Binomial Distribution and Autoregressive Correlation Structure of Order One: NB-INAR(1)*

Description

`n.nb.inar1` calculates the required sample size for proving a desired alternative when testing for a rate ratio between two groups unequal to one. Also gives back power for a specified sample size. See 'Details' for more information.

Usage

```
n.nb.inar1(alpha, power = NULL, delta, muC, size, rho, tp, k, npow = NULL,
  nmax = Inf)
```

Arguments

`alpha`	level (type I error) to which the hypothesis is tested.
`power`	power (1 - type II error) to which an alternative should be proven.
`delta`	the rate ratio which is to be proven.
`muC`	the rate observed within the control group.
`size`	dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive, need not be integer (see `rnbinom.inar1`).
`rho`	correlation coefficient of the underlying autoregressive correlation structure. Must be between 0 and 1 (see `rnbinom.inar1`).
`tp`	number of observed time points. (see `rnbinom.inar1`)
`k`	sample size allocation factor between groups: see 'Details'.
`npow`	sample size for which a power is to be calculated. Can not be specified if power is also specified.
`nmax`	maximum total sample size of both groups. If maximum is reached a warning message is broadcasted.

Details

When testing for differences between rates μ_C and μ_T of two groups, a control and a treatment group respectively, we usually test for the ratio between the two rates, i.e. $\mu_T/\mu_C = 1$. The ratio of the two rates is refered to as δ, i.e. $\delta = \mu_T/\mu_C$.

`n.nb.inar1` gives back the required sample size for the control and treatment group required to prove an existing alternative `theta` with a specified power `power` when testing the null hypothesis $H_0 : \mu_T/\mu_C \geq 1$ to level `alpha`. If `power` is not specified but instead `npow`, the power achieved with a total sample size of `npow` is calculated.

For sample sizes n_C and n_T of the control and treatment group, respectively, the argument `k` is the sample size allocation factor, i.e. $k = n_T/n_C$.

Value

rnbinom.inar1 returns the required sample size within the control group and treatment group.

Source

rnbinom.inar1 uses code contributed by Thomas Asendorf.

See Also

rnbinom.inar1 for information on the NB-INAR(1) model, fit.nb.inar1 for calculating initial parameters required when performing sample size estimation, bssr.nb.inar1 for blinded sample size reestimation within a running trial.

Examples

```
#Calculate required sample size to find significant difference with
#80% probability when testing the Nullhypothesis H_0: mu_T/mu_C >= 1
#assuming the true effect delta is 0.8 and rate, size and correlation
#parameter in the control group are 2, 1 and 0.5, respectively.

estimate<-n.nb.inar1(alpha=0.025, power=0.8, delta=0.8, muC=2, size=1, rho=0.5, tp=7, k=1)
summary(estimate)

estimate<-n.nb.inar1(alpha=0.025, npow=200, delta=0.8, muC=2, size=1, rho=0.5, tp=7, k=1)
summary(estimate)
```

r.1subgroup *Generate dataset of normal distributed observations in a one subgroup design*

Description

r.1subgroup generates data for a design with one subgroup within a full population. Each observation is normal distributed with mean 0 in the placebo group and a potential effect in the treatment group. Whether the effect is solely in the subgroup or additionally a certain amount outside of the subgroup can be specified as well as potentially different variances within the subgroup and outside of the subgroup.

Usage

```
r.1subgroup(n, delta, sigma, tau, fix.tau = c("YES", "NO"), k)
```

Arguments

n	number of observations. If length(n) > 1, the length is taken to be the number required.
delta	vector of treatment effects in the treatment group, c(outside subgroup, within subgroup).

`sigma` vector of standard deviations, c(outside subgroup, inside subgroup).

`tau` subgroup prevalence.

`fix.tau` subgroup prevalence fix or simulated according to tau, see 'Details'.

`k` sample size allocation factor between groups: see 'Details'.

Details

For `delta=` $(\Delta_F\S, \Delta_S)'$ and `sigma=` $(\sigma_F\S, \sigma_S)'$ this function `r.1subgroup` generates data as follows:

Placebo group outside of subgroup $N(0, \sigma_F^2\S)$, Placebo group within subgroup $N(0, \sigma_S^2)$, Treatment group outside of subgroup $N(\Delta_F\S, \sigma_F^2\S)$, Treatment group within subgroup $N(\Delta_S, \sigma_S^2)$.

If `fix.tau=YES` the subgroup size is generated according to the prevalence `tau`, i.e. $n_S = \tau * n$. If `fix.tau=YES`, then each new generated observations probability to belong to the subgroup is $Ber($`tau`$)$ distributed and therefore only $E(n_s) = \tau * n$ holds.

The argument `k` is the sample size allocation factor, i.e. let n_C and n_T denote the sample sizes of of the control and treatment group, respectively, then $k = n_T/n_C$.

Value

`r.1subgroup` returns a data matrix of dimension `n x 3`. The first column `TrPl` defines whether the observation belongs to the treatment group (`TrPl=0`) or to the placebo group (`TrPl=1`). Second column contains the grouping variable `FS`. For `FS=1` the observation stems from the subgroup, for `FS=0` from the full population without the subgroup. In the last column `value` the observation can be found. between time points.

Source

`r.1subgroup` uses code contributed by Marius Placzek.

Examples

```
set.seed(142)
random<-r.1subgroup(n=50, delta=c(0,1), sigma=c(1,1), tau=0.4, fix.tau="YES", k=2)
random
```

`r.gee.1subgroup` *Generate a dataset of normally distributed repeated measures in a one subgroup setting*

Description

r.gee.1subgroup generates data of a population which is comprised of a subgroup and the complementary subgroup. The generated longitudinal data needs the specification of the correlation (ρ),the correlation structure (θ) and the number of repeated measurements. The intra-subject correlation is defined via $corr(y_{ij}, y_{io}) = \rho^{(j-o)^{\theta}}$ for the correlation between timepoints i and o.The outcomes are generated as follows:

$$y_{ij} = \beta_0 + \beta_1 * I_{treat} + \beta_2 * j + \beta_3 * I_{treat} * j + \epsilon_{ij}$$

with i being the subject index and j being the time index. The regression coefficients and outcome-variance for subpopulation and complementary population can be defined seperatly .

Usage

```
r.gee.1subgroup(n, reg, sigma, rho, theta, tau, k, Time, OD)
```

Arguments

n overall sample size that is generated

reg a list containing regression coefficients for complementary population, reg[[1]] and subpopulation, reg[[2]]: see 'Details'

sigma vector of standard deviations for ϵ_{ij}, c(complementary population, subpopulation)

rho correlation between two adjacent timepoints 1 timeunit appart

theta variable specifying the type of the correlation structure: see 'Details'

tau prevalence of the subgroup in the full population.

k sample size allocation factor between control and treatment: see 'Details'.

Time list with the time-values which are taken by j: see 'Details'

OD overall dropout observed at the last timepoint in percent: see 'Details'

Details

Given the coefficients reg=list(c($\beta_0^F\S, \beta_1^F\S, \beta_2^F\S, \beta_3^F\S$), c($\beta_0^S, \beta_1^S, \beta_2^S, \beta_3^S$)) and the outcome-variance sigma=($\sigma_F\S, \sigma_S$) function r.gee.1subgroup generates data with intra-subject correlation defined by variables ρ and θ as follows:

Placebo group - complementary population $y_{ij} = \beta_0 + \beta_2 * j + N(0, \sigma_F\S)$, Placebo group - within subgroup $y_{ij} = \beta_0 + \beta_2 * j + N(0, \sigma_S)$, Treatment group - complementary population $y_{ij} = \beta_0 + \beta_1 + \beta_2 * j + \beta_3 * j + N(0, \sigma_F\S)$, Treatment group - within subgroup $y_{ij} = \beta_0 + \beta_1 + \beta_2 * j + \beta_3 * j + N(0, \sigma_S)$.

The intra-subject correlation is included by correlating the error terms ϵ_{ij}. The formula which describes the correlation between two timepoints is $corr(\epsilon_i j, \epsilon_i o) = \rho^{(j-o)^{\theta}}$. If for example $\theta = 0$ the correlation is compound symmetric. With $\theta = 0$ the data is AR(1) correlated.

Argument k is the sample size allocation factor, i.e. the ratio between control and treatment. Let n_C and n_T denote the sample sizes of of the control and treatment group, respectively, then $k = n_T/n_C$. Argument Time is a vector which are the measurment times, i. e. all the timepoint where a measurement was taken. For Time=0:5 measurments at baseline, and at timepoints 1,2,3,4 and 5 where taken.

Argument OD sets the overall dropout rate at the last timepoint. For OD=0.5 50 percent of all observation had an dropout event. If a subjact has a dropout the chance for that dropout is equally distributet over all time points.

Value

r.gee.1subgroup returns a list with 7 diffrent matrices.In every Matrix the rows are the simulated subjects and the columns are the observed time points.

The first matrix contains the id's of the subject. The id's range from 1 to N. The second are are the outcomes of a subject, y_ij, and so the dependent variable in most analysis. The outcome for y_ij can be found in row i at the corresponding collumn for j. Matrix 3 to 5 are the values for the independent variables Baseline, Gr and Time. All entries of Baseline are 1 and as such the baseline of control pations is defined by β_0. The enries of Gr corresponds to coefficient β_1, Time to coefficient β_2 and the result of Gr*Time to coefficient β_3. The sixth matrix contains the error-terms to preserve the abilety to look tat them later. The last matrix provides the invoramtion if an observation comes from an subjoct of the subpopulation or the complementary population.

Source

r.gee.1subgroup uses code contributed by Roland Gerard Gera

Examples

```
set.seed(2015)
dataset<-r.gee.1subgroup(n=200, reg=list(c(0,0,0,0.1),c(0,0,0,0.1)), sigma=c(3,2.5),
tau=0.5, rho=0.25, theta=1, k=1.5, Time=c(0:5), OD=0)
dataset
```

rnbinom.inar1 *Generate Time Series with Negative Binomial Distribution and Autoregressive Correlation Structure of Order One: NB-INAR(1)*

Description

rnbinom.inar1 generates one or more independent time series following the NB-INAR(1) model. The generated data has negative binomial marginal distribution and an autoregressive covariance structure.

Usage

```
rnbinom.inar1(n, size, mu, rho, tp)
```

Arguments

`n` number of observations. If length(n) > 1, the length is taken to be the number required.

`size` dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive, need not be integer.

`mu` parametrization via mean: see 'Details'.

`rho` correlation coefficient of the underlying autoregressive correlation structure. Must be between 0 and 1.

`tp` number of observed time points.

Details

The generated marginal negative binomial distribution with mean `mu` = μ and `size` = η has density

$$(\mu/(\mu+\eta))^x\Gamma(x+\eta)/(\Gamma(x+1)\Gamma(\eta))(\eta/(\mu+\eta))^\eta$$

for $0 < \mu$, $0 < \eta$ and $x = 0, 1, 2, ...$

Within the NB-INAR(1) model, the correlation between two time points t and s for `rho` = ρ is given through

$$\rho^{|}t - s|$$

for $0 \leq \rho \leq 1$.

Value

`rnbinom.inar1` returns a matrix of dimension `n` x `tp` with marginal negative binomial distribution with mean `mu` and dispersion parameter `size`, and an autoregressive correlation structure between time points.

Source

`rnbinom.inar1` computes a reparametrization of the NB-INAR(1) model by *McKenzie 1986* using code contributed by Thomas Asendorf.

References

McKenzie Ed (1986), Autoregressive Moving-Average Processes with Negative-Binomial and Geometric Marginal Distributions. *Advances in Applied Probability* Vol. 18, No. 3, pp. 679-705.

Examples

```
set.seed(8)
random<-rnbinom.inar1(n=1000, size=0.6, mu=2, rho=0.8, tp=6)
cor(random)

#Check the marginal distribution of time point 3
plot(table(random[,3])/1000, xlab="Probability", ylab="Observation")
lines(0:26, dnbinom(0:26, mu=2, size=0.6), col="red")
legend("topright",legend=c("Theoretical Marginal Distribution", "Observed Distribution"),
col=c("red", "black"), lty=1, lwd=c(1,2))
```

sandwich — *Estimate the Robust covariance estimator for GEE (weigthed GEE if missing occures) of Regressor parameters*

Description

`sandwich` calculates the asymptotic regression covariance structure given matrices `yCov`, `D`,`V`, `correctionmatrix` for further analyses and is a morte generalized, but also more complex version as `sandwich2`.

Usage

```
sandwich(yCov, D, V, correctionmatrix, missing = rep(0, dim(yCov)[[2]]),
  missingtype = c("none", "monotone", "intermittened"))
```

Arguments

`yCov` $yCov$ is the empirical or estimated Covariancematrix that we cen get from the outcomes. see 'Details'.

`D` D es the mean Matrix of all entries of $\Delta\mu_i/\delta\beta$, where is the average over all `i` patiens. see 'Details'.

`V` V is the Working covariance matrix. see 'Details'.

`correctionmatrix` a correctionamtrix that will correct misstakes. see 'Details' to see what these misstakes are und how to select correction matrices. see 'Details'.

`missing` a vector which describes the probabilety to experience a dropout at all observed timepoints. if missing is `"none"` then it is treated as if all entries are 0

`missingtype` describes the type of missing that occured in tzhe data. Possebiletys range from `none` if there is no missing, to `"monotone"` if missing is monotone, aka dropout, and lastly `"intermittened"` if the missingness is independent across ale timepoints

Details

`yCov` is the either empirical or estimated intra-subject covariancematrix which is needed to calculate the sandwich (robust) covariance estimator. This matrix can either be achieved by estimating the empirical intra-subject covariance out of data or by using `gen_cov_cor` to calculate a estimation for the covariance.

`D` is the estimation of $n^{-}1 * \sum_i^N \Delta\mu_i/\delta\beta$, so $D = E(D_i)$. But this is also source of an error which has to be corrected by `correctionsmatrix`. The error emerges when we calculate the "Bread" and "Meat" of the sandwichestimators. Exemplary on the "Bread" we need to calculate $E(D_i \times V \times D_i)$ wich is however enequal to $E(D_i)^t \times V \times E(D_i)$ which we ARE calculating. `correctionmatrix` is now used to correct made misstakes so that $E(D_i)^t \times V \times E(D_i)$*`correctionsmatrix`=$E(D_i \times V \times D_i)$, which is still a point which we will improve on further itterations of the algorithm.

Value

sandwich returns the sandwich (robust) covariance estimator of regression coefficients which are impicently defined by D.

Source

sandwich computes the asymptotic sandwich covariance estimator and uses code contributed by Roland Gerard Gera.

References

Liang Kung-Yee, Zeger Scott L. (1986); Jung Sin-Ho, Ahn Chul (2003); Wachtlin Daniel Kieser Meinhard (2013)

Examples

```
#Lets assume we wish to calculate the robust variance estimator for the equation
#\eqn{y_{ij}=\beta_0+\beta_1*I_{treat}+\beta_2*j+\beta_3*I _{treat}*j+\epsilon_{ij}}.
#Furthermore we use the identitiy matrix as working covariance matrix.
#We compare the results with the same estimation made by \code{sandwich2} to show the
#same results. The cance to get randomized to treatment is 60 percent and we observe
#the timerange 0:5.

  ycov = gen_cov_cor(var = 3,rho = 0.25,theta = 1,Time = 0:5,cov = TRUE)
  D = matrix(c(1,0.6,0,0,
               1,0.6,1,0.6,
               1,0.6,2,1.2,
               1,0.6,3,1.8,
               1,0.6,4,2.4,
               1,0.6,5,3.0),nrow=4)

 D=t(D)
 V=diag(1,length(0:5))
 #We correct entries where E(D_i %*% D_i) is unequal to E(D_i)%*%E(D_i) (D %*% D).
 correctionmatrix=matrix(c(1,1,1,1,1,1/0.6,1,1/0.6,1,1,1,1,1,1/0.6,1,1/0.6),nrow=4)
 missingtype = "none"

 robust=sandwich(yCov=ycov,D=D,V=V,missingtype=missingtype,correctionmatrix=correctionmatrix)
 robust

 #To see if that is correct we can verify it with function sandwich2, which is usable for
 #this particular model with:
 robust2=sandwich2(sigma = c(3,3),rho = 0.25,theta = 1,k = 1.5,Time = 0:5,
 dropout = rep(0,6),Model = 1)
 robust2[[1]]

 # We can also test this with the the Model:
 #\eqn{y_{ij}=\beta_0+\beta_2*j+\beta_3*I _{treat}*j+\epsilon_{ij}} which leads to
 D = matrix(c(1,0,0,
               1,1,0.6,
               1,2,1.2,
               1,3,1.8,
               1,4,2.4,
               1,5,3.0),nrow=3)
 D=t(D)
 V=diag(1,length(0:5))
 #We correct entries where E(D_i %*% D_i) is unequal to E(D_i)%*%E(D_i) (D %*% D).
 correctionmatrix =matrix(c(1,1,1, 1,1,1, 1,1,1/0.6),nrow=3)
 missingtype = "none"

 robust=sandwich(yCov=ycov,D=D,V=V,missingtype=missingtype,correctionmatrix=correctionmatrix)
 robust
 robust2=sandwich2(sigma = c(3,3),rho = 0.25,theta = 1,k = 1.5,Time = 0:5,
 dropout = rep(0,6),Model = 2)
 robust2[[1]]
```

sandwich2 — *Generate Time Series with Negative Binomial Distribution and Autoregressive Correlation Structure of Order One: NB-INAR(1)*

Description

rnbinom.inar1 generates one or more independent time series following the NB-INAR(1) model. The generated data has negative binomial marginal distribution and an autoregressive covariance structure.

Usage

```
sandwich2(sigma, rho, theta, k, Time, dropout, Model)
```

Arguments

sigma	assymptotic standard deviation for Full and subpupulation
rho	correlation coefficient of the underlying autoregressive correlation structure. Must be between 0 and 1.
theta	correlation absorption coefficient if tinepoints are farther appart
k	sample size allocation factor between groups: see 'Details'.
Time	vector of measured timepoints
dropout	vector describing the percentage of dropout in every timepoint
Model	either 1 or 2, describing if 4-regressor or 3-regressor model was used.

Details

The generated marginal negative binomial distribution with mean mu = μ and size = η has density

$$(\mu/(\mu+\eta))^x\Gamma(x+\eta)/(\Gamma(x+1)\Gamma(\eta))(\eta/(\mu+\eta))^\eta$$

for $0 < \mu$, $0 < \eta$ and $x = 0, 1, 2, ...$

Within the NB-INAR(1) model, the correlation between two time points t and s for `rho` = ρ is given through

$$\rho^|t - s|$$

for $0 \leq \rho \leq 1$.

Value

`rnbinom.inar1` returns a matrix of dimension `n` x `tp` with marginal negative binomial distribution with mean `mu` and dispersion parameter `size`, and an autoregressive correlation structure between time points.

Source

`rnbinom.inar1` computes a reparametrization of the NB-INAR(1) model by *McKenzie 1986* using code contributed by Thomas Asendorf.

References

McKenzie Ed (1986), Autoregressive Moving-Average Processes with Negative-Binomial and Geometric Marginal Distributions. *Advances in Applied Probability* Vol. 18, No. 3, pp. 679-705.

Examples

```
set.seed(8)
random<-rnbinom.inar1(n=1000, size=0.6, mu=2, rho=0.8, tp=6)
cor(random)

#Check the marginal distribution of time point 3
plot(table(random[,3])/1000, xlab="Probability", ylab="Observation")
lines(0:26, dnbinom(0:26, mu=2, size=0.6), col="red")
legend("topright",legend=c("Theoretical Marginal Distribution", "Observed Distribution"),
col=c("red", "black"), lty=1, lwd=c(1,2))
```

`sim.bssr.1subgroup` *Simulation of a One Subgroup Design with Internal Pilot Study*

Description

Given estimates of the treatment effects to be proven, the variances, and the prevalence, `sim.bssr.1subgroup` calculates a initial sample size and performes a blinded sample size recalculation after a prespecified number of subjects have been enrolled. Each oberservation is simulated and a final analysis executed. Several variations are included, such as different approximations or sample size allocation.

Usage

```
sim.bssr.1subgroup(nsim = 1000, alpha, beta, delta, sigma, tau, vdelta,
  vsigma, vtau, rec.at = 1/2, eps = 0.001, approx = c("conservative.t",
  "liberal.t", "normal"), df = c("n", "n1"), fix.tau = c("YES", "NO"),
  k = 1, adjust = c("YES", "NO"))
```

Arguments

`nsim`	number of simulation runs.
`alpha`	level (type I error) to which the hypothesis is tested.
`beta`	type II error (power=1-beta) to which an alternative should be proven.
`delta`	vector of true treatment effects, c(outside subgroup, inside subgroup).
`sigma`	vector of true standard deviations, c(outside subgroup, inside subgroup).
`tau`	subgroup prevalence.
`vdelta`	vector of treatment effects to be proven, c(outside subgroup, inside subgroup).
`vsigma`	vector of assumed standard deviations, c(outside subgroup, inside subgroup).
`vtau`	expected subgroup prevalence.
`rec.at`	blinded sample size review is performed after `rec.at`*100% subjects of the initial sample size calculation.
`eps`	precision parameter concerning the power calculation in the iterative sample size search algorithm.
`approx`	approximation method: Use a conservative multivariate t distribution ("conservative.t"), a liberal multivariate t distribution ("liberal.t") or a multivariate normal distribution ("normal") to approximate the joint distribution of the standardized teststatistics.
`df`	in case of a multivariate t distribution approximation, recalculate sample size with degrees of freedom depending on the size of the IPS (df=n1) or depending on the final sample size (df=n).
`fix.tau`	subgroup prevalence is fixed by design (e.g. determined by recruitment) or is simulated and has to be reestimated during the blinded review.
`k`	sample size allocation factor between groups: see 'Details'.
`adjust`	adjust blinded estimators for assumed treatment effect ("YES","No").

Details

This function combines sample size estimation, blinded sample size reestimation and analysis in a design with a subgroup within a full population where we want to test for treatment effects between a control and a treatment group. The required sample size for the control and treatment group to prove an existing alternative `delta` with a specified power 1-`beta` when testing the global null hypothesis $H_0 : \Delta_F = \Delta_S = 0$ to level `alpha` is calculated prior to the study and then recalculated in an internal pilot study.

For sample sizes n_C and n_T of the control and treatment group, respectively, the argument `k` is the sample size allocation factor, i.e. $k = n_T/n_C$.

The parameter df provides a difference to the standard sample size calculation procedure implemented in n.1subgroup. When applying a multivariate t distribution approximation to approximate the joint distribution of the standardized test statistics it gives the opportunity to use degrees of freedom depending on the number of subjects in the IPS instead of degrees of freedom depending on the projected final sample size. Note that this leads to better performance when dealing with extremely small subgroup sample sizes but significantly increases the calculated final sample size.

Value

sim.bssr.1subgroup returns a data.frame containing the mean recalculated sample size within the control group and treatment group and the achieved simulated power along with all relevant parameters.

Source

sim.bssr.1subgroup uses code contributed by Marius Placzek.

See Also

sim.bssr.1subgroup makes use of n.1subgroup, bssr.1subgroup, and r.1subgroup.

Examples

```
sim.bssr.1subgroup(nsim=10,alpha=0.025,beta=0.1,delta=c(0,1),sigma=c(1,1.3),tau=0.2,
vdelta=c(0,1),vsigma=c(1,1),vtau=0.3,eps=0.002, approx="conservative.t",df="n",
fix.tau="YES",k=1,adjust="NO")
```

---

sim.bssr.gee.1subgroup

Simulation of a longitudinal One Subgroup Design with Internal Pilot Study

---

Description

Given estimates of the treatment effects to be proven, the variances, and the prevalence, sim.bssr.gee.1subgroup calculates an initial sample size and performes a blinded sample size recalculation after a prespecified number of subjects have been enrolled. Each oberservation is simulated and a final analysis executed. Several variations are included, such as different approximations or sample size allocation.

Usage

```
sim.bssr.gee.1subgroup(nsim = 1000, alpha = 0.05, tail = "both",
  beta = 0.2, delta = c(0.1, 0.1), vdelta = c(0.1, 0.1),
  sigma_pop = c(3, 3), vsigma_pop = c(3, 3), tau = 0.5, rho = 0.25,
  vrho = 0.25, theta = 1, vtheta = 1, Time = 0:5, rec.at = 0.5,
  k = 1, model = 1, V = diag(rep(1, length(Time))), OD = 0,
  vdropout = rep(0, length(Time)), missingtype = "none",
  vmissingtype = "none", seed = 2015)
```

Arguments

`nsim` number of simulation runs.

`alpha` level (type I error) to which the hypothesis is tested.

`tail` which type of test is used, e.g. which quartile und H0 is calculated

`beta` type II error (power=1-beta) to which an alternative should be proven.

`delta` vector of true treatment effects, c(allcomers, inside subgroup).

`vdelta` vector of treatment effects to be proven, c(allcomers, inside subgroup).

`sigma_pop` vector of true standard deviations, c(allcomers, inside subgroup).

`vsigma_pop` vector of assumed standard deviations, c(allcomers, inside subgroup).

`tau` subgroup prevalence.

`rho` true correlation coefficent between two adjasent timepints

`vrho` initial expectation of the correlation coefficent between two adjasent timepints

`theta` true correlation absorption coefficient if tinepoints are farther appart

`vtheta` expected correlation absorption coefficient if tinepoints are farther appart

`Time` vector of measured timepoints

`rec.at` blinded sample size review is performed after `rec.at`*100% subjects of the initial sample size calculation.

`k` sample size allocation factor between groups: see 'Details'.

`model` which of the two often revered statistical models should be used?: see 'Details'.

`V` working covariance matrix. Easiest case ist the identity matrix.

`OD` overall dropout measuered at last timepoint

`vdropout` vector of expected dropouts per timepoint if missingness is to be expected

`missingtype` true missingtype underlying the missingness

`vmissingtype` initial assumptions about the missingtype underlying the missingness

`seed` set seed value for the simulations to compare resutlts.

Details

This function combines sample size estimation, blinded sample size reestimation and analysis in a design with a subgroup within a full population where we want to test for treatment effects between a control and a treatment group. The required sample size for the control and treatment group to prove an existing alternative `delta` with a specified power 1-`beta` when testing the global null hypothesis $H_0 : \Delta_F = \Delta_S = 0$ to level `alpha` is calculated prior to the study and then recalculated in an internal pilot study.

For sample sizes n_C and n_T of the control and treatment group, respectively, the argument `k` is the sample size allocation factor, i.e. $k = n_T/n_C$.

Value

`sim.bssr.1subgroup` returns a data.frame containing the mean and variance of recalculated sample sizes within the control group and treatment group respectively and the achieved simulated power along with all relevant parameters.

Source

sim.bssr.gee.1subgroup uses code contributed by Roland Gerard Gera.

See Also

sim.bssr.gee.1subgroup makes use of n.gee.1subgroup, bssr.gee.1subgroup, and r.gee.1subgroup.

Examples

```
sim.bssr.gee.1subgroup(nsim = 5,missingtype = "intermittened")
```

---

summary.bssrest *Summarizing Blinded Sample Size Reestimation*

Description

summary method for class "bssrest".

Usage

```
## S3 method for class 'bssrest'
summary(object, ...)
```

Arguments

object	an object of class "bssrest".
...	Arguments to be passed to methods.

Details

summary.bssrest gives back blinded sample size estimates. Furthermore, inputs are displayed for double checking.

See Also

n.nb.inar1 for initial sample size estimates within the NB-INAR(1) model.

Examples

```
#Calculate required sample size to find significant difference with
#80% probability when testing the Nullhypothesis H_0: mu_T/mu_C >= 1
#assuming the true effect delta is 0.8 and rate, size and correlation
#parameter in the control group are 2, 1 and 0.5, respectively.

estimate<-n.nb.inar1(alpha=0.025, power=0.8, delta=0.8, muC=2, size=1, rho=0.5, tp=7, k=1)

#Simulate data
set.seed(8)
```

```
placebo<-rnbinom.inar1(n=50, size=1, mu=2, rho=0.5, tp=7)
treatment<-rnbinom.inar1(n=50, size=1, mu=1.6, rho=0.5, tp=7)

#Blinded sample size reestimation
estimate<-bssr.nb.inar1(alpha=0.025, power=0.8, delta=0.8, x=rbind(placebo, treatment),
n=c(50,50), k=1)
summary(estimate)
```

---

summary.ssest *Summarizing Initial Sample Size Estimates*

Description

summary method for class "ssest".

Usage

```
## S3 method for class 'ssest'
summary(object, ...)
```

Arguments

object	an object of class "ssest".
...	Arguments to be passed to methods.

Details

summary.ssest gives back initial sample size estimates required. Furthermore, inputs are displayed for double checking.

See Also

n.nb.inar1 for initial sample size estimates within the NB-INAR(1) model.

Examples

```
#Calculate required sample size to find significant difference with
#80% probability when testing the Nullhypothesis H_0: mu_T/mu_C >= 1
#assuming the true effect delta is 0.8 and rate, size and correlation
#parameter in the control group are 2, 1 and 0.5, respectively.

estimate<-n.nb.inar1(alpha=0.025, power=0.8, delta=0.8, muC=2, size=1, rho=0.5, tp=7, k=1)
summary(estimate)
```

Index