Package 'spass'

July 23, 2025

Type Package

Title Study Planning and Adaptation of Sample Size

Version 1.3

Date 2020-12-01

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.

License GPL

Encoding UTF-8

LazyData true

Imports Rcpp,mvtnorm,multcomp,MASS,geepack,stats,utils

LinkingTo Rcpp

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-12-07 22:40:02 UTC

Contents

`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 re-estimations from an Internal Pilot Study (IPS), `bssr.GEE.1subgroup` re-estimates required sample size given the re-estimated nuisance parameters are given. `bssr.gee.1subgroup` is a wrapper for `n.gee.1subgroup` where the re-estimation of the variances can be highly dependable on the user and should be supplied separately. see "detail" for more information.

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 estimated treatment effect in overall and sub population, c(overall population, only subpopulation).
`estsigma`	vector of re-estimated standard deviations, c(full population, subpopulation). See 'Details'.
`tau`	ratio between complementary F/S and sub-population S.
`k`	treatment allocation factor between groups: see 'Details'.

Details

This function provides a simple warped for `n.gee.1subgroup` where instead of initial assumptions, reestimated nuisance parameter are used. For more information see `n.gee.1subgroup`. Required samplesize to test alternative `delta` with specified power 1-`beta` when testing the global null hypothesis $H_0 : \beta_3^F = \beta_3^S = 0$ to level `alpha` is estimated. When testing outcomes have variance `estsigma`.

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. $k = n_T/n_C$ and `tau` represents the ratio of the sub-population.

Value

`bssr.gee.1subgroup` returns a list containing the recalculated sample sizes 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 a trial and `estimcov` how the re-estimate nuisance parameters. See $sim.gee$ for a working example for an initial sample size estimation and a re-estimation mid trial.

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.gf` *Blinded Sample Size Reestimation for Longitudinal Count Data with marginal Negative Binomial Distribution and underlying Gamma Frailty with Autoregressive Correlation Structure of Order One*

Description

`bssr.nb.gf` fits blinded observations and recalculates the sample size required for sustaining power at desired alternative when testing for trend parameters in a Gamma frailty models. See 'Details' for more information.

Usage

```
bssr.nb.gf(
  data,
  alpha = 0.025,
  power = 0.8,
  delta,
```

```
  h0 = 0,
  tp,
  k,
  trend = c("constant", "exponential", "custom"),
  approx = 20
)
```

Arguments

data	a matrix or data frame containing count data which is to be fitted. Columns correspond to time points, rows to observations.
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 relevant effect size, which is assumed to be true, see 'Details'.
h0	the value against which h is tested, see 'Details'.
tp	number of observed time points. (see rnbinom.gf)
k	sample size allocation factor between groups: see 'Details'.
trend	the trend which assumed to underlying in the data.
approx	numer of iterations in numerical calculation of the sandwich estimator, see 'Details'.

Details

The function recalculates a sample size for testing in constant and exponential trends.

Under a constant trend, the means in control and experiment group are equal to λ_1 and $\lambda_1 + \lambda_2$, respectively. The treatment effect delta is therefore equal to λ_2.

Under an exponential trend, the means in control and experiment group are equal to $exp(\lambda_1 + t \cdot \lambda_2)$ and $\lambda_1 + t \cdot \lambda_2 + t \cdot \lambda_3$, respectively. The treatment effect delta is therefore equal to λ_3.

bssr.nb.gf returns the required sample size for the control and treatment group required to prove an existing alternative delta with a specified power power when testing the null hypothesis H_0 : $\delta \geq h_0$ at level alpha. Nuisance parameters are estimated through the blinded observations data, 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

bssr.nb.gf returns the required sample size within the control group and treatment group.

Source

bssr.nb.gf uses code contributed by Thomas Asendorf.

See Also

rnbinom.gf for information on the Gamma Frailty model, n.nb.gf for calculating initial sample size required when performing inference, fit.nb.gf for calculating initial parameters required when performing sample size estimation.

Examples

```
##The example is commented as it may take longer than 10 seconds to run.
##Please uncomment prior to execution.

##Example for constant rates
#set.seed(12)
#h<-function(lambda.eta){
#   lambda.eta[2]
#}
#hgrad<-function(lambda.eta){
#   c(0, 1, 0)
#}

##Calculate initial sample size
#estimate<-n.nb.gf(lambda=c(0,-0.3), size=1, rho=0.5, tp=6, k=1, h=h, hgrad=hgrad,
#   h0=0, trend="constant", approx=20)

##Generate and permutate data with different nuisance parameters
#random<-get.groups(n=round(estimate$n/2), size=c(0.8, 0.8), lambda=c(0.5, -0.3),
#   rho=c(0.4, 0.4), tp=6, trend="constant")
#random<-random[sample(1:nrow(random), nrow(random)), ]

##Recalculate sample size with data
#reestimate<-bssr.nb.gf(data=random, alpha=0.025, power=0.8, delta=-0.3, h0=0,
#   tp=6, k=1, trend="constant", approx = 20)

#summary(reestimate)
```

`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 simulation parameters*

Description

`estimcov` estimates the covariance matrix and dropout rates given a dataset and observation-times

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`	matrix with the dataset which is used to estimate the covariance and dropout structure.
`Time`	vector with observation-times.
`Startvalues`	vector with starting values for variance, `rho` and `theta` respectively.
`stepwidth`	vector describing the step length of previously mentioned values.
`maxiter`	maximum amount of iterations
`lower`	vector with minimum for the parameters described in Startvalues
`upper`	vector with maximum for the parameters described in Startvalues

Details

This function is designed to estimate the variance, `rho` and `theta` and a vector with the dropout rate in the data.

Value

`estimcov` returns a list with two entries. In the first the parameters variance, `rho` and `theta` are returned and in the second a vector with the dropout-rate is returned.

Source

`estimcov` uses code contributed by Roland Gerard Gera.

Examples

```
# First generate a dataset with 200 patients, rho =0.25 and tau = 0.5 and
# then estimate the parameters using estimcov.

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)

estimations <- estimcov(data=dataset,Time=c(0:5))
estimations[[1]]
estimations[[2]]
```

`fit.nb.gf` *Fitting Longitudinal Data from a Gamma Frailty Model with Frailty of Autoregressive Correlation Structure of Order One*

Description

`fit.nb.gf` fits data using the pseudo maximum likelihood of a Gamma frailty model

Usage

```
fit.nb.gf(
  dataC,
  dataE,
  trend = c("constant", "exponential"),
  lower,
  upper,
  method = "L-BFGS-B",
  start,
  approx = 20,
  rho = FALSE,
  H0 = FALSE,
  h0 = 0
)
```

Arguments

`dataC`	a matrix containing count data from the control group, which is to be fitted. Columns correspond to time points, rows to observations.
`dataE`	a matrix containing count data from the experiment group, which is to be fitted. Columns correspond to time points, rows to observations.
`trend`	the trend which assumed to underlying in the data.
`lower`	vector of lower bounds for estimated parameters `lambda`, `size` and `rho`, respectively.

upper	vector of upper bounds for estimated parameters lambda, 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.
approx	numer of iterations in numerical calculation of the sandwich estimator, see 'Details'.
rho	indicates whether or not to calculate the correlation coefficient of Gamma frailties. Must be TRUE or FALSE.
H0	indicates whether or not to calculate the hessian and outer gradient matrix under the null hypothesis, see 'Details'.
h0	the value against which is tested under the null

Details

the function fit.nb.gf fits a Gamma frailty model as found in Fiocco (2009). The fitting function allows for incomplete follow up, but not for intermittent missingness.

When calculating the expected sandwich estimator required for the sample size, certain terms can not be computed analytically and have to be approximated numerically. The value approx defines how close the approximation is to the true expected sandwich estimator. High values of approx provide better approximations but are compuationally more expensive.

If parameter H0 is set to TRUE, the hessian and outer gradient are calculated under the assumption that lambda[2] $\geq$ h0 if trend = "constant" or lambda[3] $\geq$ h0 if trend = "exponential".

Value

fit.nb.gf returns estimates of the trend parameters lambda, dispersion parameter size, Hessian matrix hessian, outer gradient product matrix ogradient and, if inquired, correlation coefficient rho.

Source

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

References

Fiocco M, Putter H, Van Houwelingen JC, (2009), A new serially correlated gamma-frailty process for longitudinal count data *Biostatistics* Vol. 10, No. 2, pp. 245-257.

See Also

rnbinom.gf for information on the Gamma frailty model, n.nb.gf for calculating initial sample size required when performing inference, bssr.nb.gf for blinded sample size reestimation within a running trial, optim for more information on the used minimization algorithms.

Examples

```
#Generate data from the Gamma frailty model
random<-get.groups(n=c(1000,1000), size=c(0.7, 0.7), lambda=c(0.8, -0.5), rho=c(0.6, 0.6),
  tp=7, trend="constant")
fit.nb.gf(dataC=random[1001:2000,], dataE=random[1:1000,], trend="constant")
```

`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 a covariance or a correlation matrix*

Description

Generate a covariance or correlation matrix given 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 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

gen_cov_cor is used to generate either a covariance or a correlation matrix. Given vector Time and parameters var, rho and theta 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)]]

Value

gen_cov_cor returns a covariance or correlation matrix.

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 re-estimation within a trial. See estimcov for more information on the used minimization algorithms.

Examples

```
#Generate a covariance-matrix with measurements at Baseline and at times c(1,1.5,2,5)

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 with the same values

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

---

get.groups *Generate Time Series with Negative Binomial Distribution and Multivariate Gamma Frailty with Autoregressive Correlation Structure of Order One with Trend*

Description

rnbinom.gf generates one or more independent time series following the Gamma frailty model. The generated data has negative binomial marginal distribution and the underlying multivariate Gamma frailty an autoregressive covariance structure.

Usage

```
get.groups(n, size, lambda, rho, tp, trend)
```

Arguments

n	number of observations.
size	dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive, need not be integer.
lambda	vector of means of trend parameters.
rho	correlation coefficient of the underlying autoregressive Gamma frailty. Must be between 0 and 1.
tp	number of observed time points.
trend	a string giving the trend which is to be simulated.

Details

The function relies on rnbinom.gf for creating data with underlying constant or exponential trends.

Value

get.groups returns a matrix of dimension n x tp with marginal negative binomial distribution with means corresponding to trend parameters lambda, common dispersion parameter size and a correlation induce by rho, the correlation coefficient of the autoregressive multivariate Gamma frailty.

Source

rnbinom.gf computes observations from a Gamma frailty model by *Fiocco et. al. 2009* using code contributed by Thomas Asendorf.

References

Fiocco M, Putter H, Van Houwelingen JC, (2009), A new serially correlated gamma-frailty process for longitudinal count data *Biostatistics* Vol. 10, No. 2, pp. 245-257.

See Also

rnbinom.gf for information on the Gamma frailty model.

Examples

```
random<-get.groups(n=c(1000,1000), size=c(0.5, 0.5), lambda=c(1, 2), rho=c(0.6, 0.6), tp=7,
  trend="constant")
head(random)
```

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*

Description

`n.gee.1subgroup` calculates the required sample size for proving a desired alternative when testing a regression coefficients in a full and/or a 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 estimated treatment effect in overall and sub population, c(overall population, only subpopulation).
sigma	vector of estimated standard deviations, 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 a sample size estimation in a design with a nested subgroup within an overall population. To calculate the required sample only the value of tested regressor needs to inserted as delta. sigma is the variance of that regressor. 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.

Here argument k denotes the sample size allocation factor between treatment groups, 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 re-estimation within a running trial and sandwich for estimating asymptotic covarianc mtrices in GEE models.

Examples

```
#Calculate required sample size to correctly reject Null with
#80% probability when testing global Nullhypothesis H_0: Delta_F=Delta_S = 0, while
#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 variances of regressors in delta when variances 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)

#Alternatively we can estimate the power our study would have
#if we know the effect in and outside our subgroup as
#well as the variance of the regressors. Here we
#estimate that only 300 Patiens total can be recruited and we are interested
#in the power that would give us.

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.gf — *Sample Size Calculation for Comparing Two Groups when observing Longitudinal Count Data with marginal Negative Binomial Distribution and underlying Gamma Frailty with Autoregressive Correlation Structure of Order One*

Description

n.nb.gf calculates required sample sizes for testing trend parameters in a Gamma frailty model

Usage

```
n.nb.gf(
  alpha = 0.025,
  power = 0.8,
  lambda,
  size,
  rho,
  tp,
  k = 1,
  h,
  hgrad,
  h0,
  trend = c("constant", "exponential", "custom"),
```

```
  approx = 20
)
```

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.
lambda	the set of trend parameters assumed to be true at the beginning prior to trial onset
size	dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive, need not be integer (see rnbinom.gf).
rho	correlation coefficient of the autoregressive correlation structure of the underlying Gamma frailty. Must be between 0 and 1 (see rnbinom.gf).
tp	number of observed time points. (see rnbinom.gf)
k	sample size allocation factor between groups: see 'Details'.
h	hypothesis to be tested. The function must return a single value when evaluated on lambda.
hgrad	gradient of function h
h0	the value against which h is tested, see 'Details'.
trend	the trend which assumed to underlying in the data.
approx	numer of iterations in numerical calculation of the sandwich estimator, see 'Details'.

Details

The function calculates required samples sizes for testing trend parameters of trends in longitudinal negative binomial data. The underlying one-sided null-hypothesis is defined by $H_0 : h(\eta, \lambda) \geq h_0$ vs. the alternative $H_A : h(\eta, \lambda) < h_0$. For testing these hypothesis, the program therefore requires a function h and a value h0.

n.nb.gf gives back the required sample size for the control and treatment group, to prove an existing alternative $h(\eta, \lambda) - h_0$ with a power of power when testing at 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. $k = n_T/n_C$.

When calculating the expected sandwich estimator required for the sample size, certain terms can not be computed analytically and have to be approximated numerically. The value approx defines how close the approximation is to the true expected sandwich estimator. High values of approx provide better approximations but are compuationally more expensive.

Value

n.nb.gf returns the required sample size within the control group and treatment group.

Source

n.nb.gf uses code contributed by Thomas Asendorf.

See Also

rnbinom.gf for information on the Gamma frailty model, fit.nb.gf for calculating initial parameters required when performing sample size estimation, bssr.nb.gf for blinded sample size reestimation within a running trial.

Examples

```
##The example is commented as it may take longer than 10 seconds to run.
##Please uncomment prior to execution.

##Example for constant rates
#h<-function(lambda.eta){
#   lambda.eta[2]
#}
#hgrad<-function(lambda.eta){
#   c(0, 1, 0)
#}

##We assume the rate in the control group to be exp(lambda[1]) = exp(0) and an
##effect of lambda[2] = -0.3. The \code{size} is assumed to be 1 and the correlation
##coefficient \code{\rho} 0.5. At the end of the study, we would like to test
##the treatment effect specified in lambda[2], and therefore define function
##\code{h} and value \code{h0} accordingly.

#estimate<-n.nb.gf(lambda=c(0,-0.3), size=1, rho=1, tp=6, k=1, h=h, hgrad=hgrad,
#   h0=0.2, trend="constant", approx=20)
#summary(estimate)

##Example for exponential trend
#h<-function(lambda.eta){
#   lambda.eta[3]
#}
#hgrad<-function(lambda.eta){
#   c(0, 0, 1, 0)
#}

#estimate<-n.nb.gf(lambda=c(0, 0, -0.3/6), size=1, rho=0.5, tp=7, k=1, h=h, hgrad=hgrad,
#   h0=0, trend="exponential", approx=20)
#summary(estimate)
```

---

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 $Ber(\text{tau})$ 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 dataset of normal distributed repeated observations in a one subgroup design*

Description

`r.gee.1subgroup` generates data for a design with one subgroup within a full population. Each baseline-observation is normal distributed with mean

$$\beta_0$$

in placebo group and

$$\beta_0 + \beta_1$$

in treatment group. Measurements after baseline have mean

$$\beta_0 + \beta_2 * t$$

in placebo group and

$$\beta_0 + \beta_1 + \beta_2 * t + \beta_3 * t$$

in treatment group where

$$t$$

is the measurement time. Whether the effect can be found solely in the subgroup or additionally a certain amount outside of the subgroup can be specified as well as a potential different covariance-structure within subgroup and in the complementary subgroup.

Usage

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

Arguments

`n`	overall sample size for the overall population
`reg`	list containing coefficients β_0 to β_0 for complementary population, `reg[[1]]` and subpopulation, `reg[[2]]`: see 'Details'.
`sigma`	vector with standard deviations for generated observations c(complementary population, subpopulation).
`rho`	variable used together with `theta` to describe correlation between two adjacent timepoints: see 'Details'.
`theta`	variable used together with `rho` to describe correlation between two adjacent timepoints: see 'Details'.

tau	subgroup prevalence.
k	sample size allocation factor between treatment groups: see 'Details'.
Time	list of timepoints t that have to be generated: see 'Details'.
OD	percentage of observed overall dropout at last timepoint: see 'Details'.

Details

For reglist(c(β_0^F§, β_1^F§, β_2^F§, β_3^F§), c(β_0^S, β_1^S, β_2^S, β_3^S)) and variances sigma=(σ_F§, σ_S) function r.gee.1subgroup generates data given correlation-variables ρ and θ as follows (and let t=0 be the baseline measurement):

Placebo group - complementary population $y_{it} = N(\beta_0 + \beta_2 * t, \sigma_F$§), Placebo group - within subgroup $y_{it} = N(\beta_0 + \beta_2 * t, \sigma_S)$, Treatment group - complementary population $y_{it} = N(\beta_0 + \beta_1 + \beta_2 * t + \beta_3 * t, \sigma_F$§), Treatment group - within subgroup $y_{it} = N(\beta_0 + \beta_1 + \beta_2 * t + \beta_3 * t, \sigma_S)$. Correlation between measurements - $corr(\epsilon_i t, \epsilon_i o) = \rho^{(t-o)^\theta}$

Argument k is the sample size allocation factor, i.e. the ratio between control and treatment. Let n_C and n_T denote sample sizes of control and treatment groups respectively, then $k = n_T/n_C$.

Argument Time is the vector denoting all measuring-times, i. e. every value for t.

Argument OD sets the overall dropout rate observed at the last timepoint. For OD=0.5, 50 percent of all observation had a dropout event at some point. If a subject experienced a dropout the starting time of the dropout is equally distributed over all timepoints.

Value

r.gee.1subgroup returns a list with 7 different entries. Every Matrix rows are the simulated subjects and the columns are the observed time points.

The first list element is a vector containing subject ids. The second element contains a matrix with the outcomes of a subject with row being the subjects and columns being the measuring-timepoints Elements 3 to 5 return matrices with the information of which patients have baseline-measurements, which patients belong to treatment and which to control and what are the observed timepoints for each patient respectively. The sixth entry returns a matrix which contains the residuals of each measurement. The seventh entry returns the sub-population identification.

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.gf` *Generate Time Series with Negative Binomial Distribution and Multivariate Gamma Frailty with Autoregressive Correlation Structure of Order One*

Description

`rnbinom.gf` generates one or more independent time series following the Gamma frailty model. The generated data has negative binomial marginal distribution and the underlying multivariate Gamma frailty an autoregressive covariance structure.

Usage

```
rnbinom.gf(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`	vector of means of time points: see 'Details'.
`rho`	correlation coefficient of the underlying autoregressive Gamma frailty. 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 $x = 0, 1, 2, ...$ Hereby, each entry of vector `mu` corresponds to one time point. Therefore, each timepoint can have its distinct mean.

Within the Gamma frailty model, the correlation between two frailties of time points t and s for `rho` = ρ is given by

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

for $0 \leq \rho \leq 1$. Note: this does not correspond to the correlation of observations.

Value

`rnbinom.gf` returns a matrix of dimension `n` x `tp` with marginal negative binomial distribution with means `mu`, common dispersion parameter `size` and a correlation induce by the autoregressive multivariate Gamma frailty.

Source

rnbinom.gf computes observations from a Gamma frailty model by *Fiocco et. al. 2009* using code contributed by Thomas Asendorf.

References

Fiocco M, Putter H, Van Houwelingen JC, (2009), A new serially correlated gamma-frailty process for longitudinal count data *Biostatistics* Vol. 10, No. 2, pp. 245-257.

Examples

```
set.seed(8)
random<-rnbinom.gf(n=1000, size=0.6, mu=1:6, 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=3, size=0.6), col="red")
legend("topright",legend=c("Theoretical Marginal Distribution", "Observed Distribution"),
  col=c("red", "black"), lty=1, lwd=c(1,2))
```

---

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 *Calculate the robust covariance estimator for GEE given an*

Description

sandwich calculates the covariance structure between timepoints given matrices yCov, D,V and correctionmatrix. This is done to be able to account for missingness in the Data.

Usage

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

Arguments

yCov	$yCov$ matrix containing either an estimation for the covariance between timepoints or an empirical covariance matrix itsel. see 'Details'.
D	D denotes the mean matrix of all entries of $\Delta\mu_i/\delta\beta$, where this is the average over all i patiens. see 'Details'.
V	V denotes the working covariance matrix. see 'Details'.
correctionmatrix	As of this version this matrix is needed to correct some calculations. see 'Details' to see for more details and how to correctly select matrices.
missing	vector which denotes the probability to experience a dropout at each timepoint. If missingtype is "none" then all entries are 0.
missingtype	String which describes the type of missingness occuring in the data. none if no missingnes occured, "monotone" if missing was monotone and "intermittened" if the missingness was independent across all timepoints.

Details

yCov is either empirical or the estimated covariance-matrix between timepoints which is needed to calculate the sandwich estimator. This matrix can either be generated by estimating the empirical covariance matrix using existing data or by using function gen_cov_cor to calculate a estimation for the covariance.

D denotes the estimation of $n^-1 * \sum_i^N \Delta\mu_i/\delta\beta$, which means that $D = E(D_i)$. As of yet this has the unfortunate side effect that E(D_i

Value

sandwich returns the robust covariance estimator of regression coefficients which are implicitly 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

```
#Let's assume we wish to calculate the robust variance estimator for  equation
#\eqn{y_{it}=\beta_0+\beta_1*I_{treat}+\beta_2*t+\beta_3*I _{treat}*t+\epsilon_{it}}.
#Furthermore we use the identitiy matrix as the working covariance matrix.
#The chance to get treatment is 60 percent and the observed timerange ranges from 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
```

`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 performs a blinded sample size recalculation after a pre-specified number of subjects have been enrolled. Each observation 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(overall population, inside subgroup).
vdelta	vector of treatment effects to be proven, c(overall population, inside subgroup).
sigma_pop	vector of true standard deviations of the treatment effects, c(overall population, subgroup).
vsigma_pop	vector of assumed standard deviations, c(overall population, inside subgroup).
tau	subgroup prevalence.
rho	true correlation coefficient between two adjacent timepoints
vrho	initial expectation of the correlation coefficient between two adjacent timepoints
theta	true correlation absorption coefficient if timepoints are farther apart
vtheta	expected correlation absorption coefficient if timepoints are farther apart
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.
OD	overall dropout measured 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 results.

Details

This function combines sample size estimation, blinded sample size re-estimation 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 or from other 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 or from other 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)
```

test.nb.gf *Testing Hypotheses in Gamma Frailty models*

Description

test.nb.gf tests hypotheses for certain trends in Gamma frailty models

Usage

```
test.nb.gf(
  dataC,
  dataE,
  h,
  hgrad,
  h0 = 0,
  trend = c("constant", "exponential", "custom"),
  H0 = FALSE,
  one.sided = TRUE,
  ...
)
```

Arguments

dataC	a matrix or data frame containing count data from the control group. Columns correspond to time points, rows to observations.
dataE	a matrix or data frame containing count data from the experiment group. Columns correspond to time points, rows to observations.
h	hypothesis to be tested. The function must return a single value when evaluated on lambda.
hgrad	gradient of function h
h0	the value against which h is tested, see 'Details'.
trend	the trend which assumed to be underlying in the data.
H0	indicates if the sandwich estimator is calculated under the null hypothesis or alternative.
one.sided	indicates if the hypothesis should be tested one- or two-sided
...	Arguments to be passed to function fit.nb.gf().

Details

the function test.nb.gf tests for the null hypothesis $h(\eta, \lambda) = h_0$ against the alternative $h(\eta, \lambda) \neq h_0$. The fitting function allows for incomplete follow up, but not for intermittent missingness.

If parameter H0 is set to TRUE, the hessian and outer gradient are calculated under the assumption that lambda[2] $\geq$ h0 if trend = "constant" or lambda[3] $\geq$ h0 if trend = "exponential".

Value

`test.nb.gf` returns effect size, standard error, Z-statistic and p-value attained through standard normal approximation.

Source

`test.nb.gf` uses code contributed by Thomas Asendorf.

References

Fiocco M, Putter H, Van Houwelingen JC, (2009), A new serially correlated gamma-frailty process for longitudinal count data *Biostatistics* Vol. 10, No. 2, pp. 245-257.

See Also

`rnbinom.gf` for information on the Gamma Frailty model, `n.nb.gf` for calculating initial sample size required when performing inference, `fit.nb.gf` for calculating initial parameters required when performing sample size estimation.

Examples

```
#Create data from two groups
random<-get.groups(n=c(100,100), size=c(0.7, 0.7), lambda=c(0.8, 0), rho=c(0.6, 0.6),
  tp=7, trend="constant")

#Define hypothesis
h<-function(lambda.eta){
  lambda.eta[2]
}
hgrad<-function(lambda.eta){
  c(0, 1, 0)
}
test.nb.gf(dataC=random[101:200,], dataE=random[1:100,], h=h, hgrad=hgrad, h0=0,
  trend="constant", H0=FALSE)
```

---

test.nb.inar1 *Testing Hypotheses in NB-INAR(1) model*

Description

`test.nb.inar1` tests hypotheses for rate ratios of two groups in an NB-INAR(1) model

Usage

```
test.nb.inar1(dataC, dataE, h0 = 1)
```

Arguments

dataC	a matrix or data frame containing count data from the control group. Columns correspond to time points, rows to observations.
dataE	a matrix or data frame containing count data from the experiment group. Columns correspond to time points, rows to observations.
h0	the value against which h is tested, see 'Details'.

Details

the function test.nb.inar1 tests for the null hypothesis $\lambda_T/\lambda_C = h0$ against the alternative $\lambda_T/\lambda_C \neq h_0$. For attaining estimates, method of moments estimators are used.

Value

test.nb.inar1 returns effect size, standard error, Z-statistic and p-value attained through standard normal approximation.

Source

test.nb.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

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

test.nb.inar1(dataC=groupC, dataE=groupE, h0=1)
```

Index