Package 'longpower'

April 20, 2021

Type Package

Title Sample Size Calculations for Longitudinal Data

Version 1.0.23

Date 2021-04-19

Description Compute power and sample size for linear models of longitudinal data. Supported models include mixed-effects models and models fit by generalized least squares and generalized estimating equations. Relevant formulas are derived by Liu and Liang (1997) <DOI:10.2307/2533554>, Diggle et al (2002) <ISBN:9780199676750>, and Lu, Luo, and Chen (2008) <DOI:10.2202/1557-4679.1098>.

License GPL (>= 2)

Depends R (>= 3.0.0), lme4 (>= 1.0), nlme

Suggests gee, testthat, methods, knitr, rmarkdown

LazyLoad yes

VignetteBuilder knitr

URL https://github.com/mcdonohue/longpower

Collate 'longpower-package.R' 'diggle.linear.power.R' 'edland.linear.power.R' 'liu.liang.linear.power.R' 'lmmpower.R' 'power_mmrm.R' 'print.power.longtest.R'

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Michael C. Donohue [aut, cre], Steve D. Edland [ctb]

Maintainer Michael C. Donohue <mdonohue@usc.edu>

Repository CRAN

Date/Publication 2021-04-20 05:40:06 UTC

R topics documented:

`longpower-package` *Sample size calculations for longitudinal data*

Description

The longpower package contains functions for computing power and sample size for linear models of longitudinal data based on the formula due to Liu and Liang (1997) and Diggle et al (1994). Either formula is expressed in terms of marginal model or Generalized Estimating Equations (GEE) parameters. This package contains functions which translate pilot mixed effect model parameters (e.g. random intercept and/or slope) into marginal model parameters so that the formulas of Diggle et al or Liu and Liang formula can be applied to produce sample size calculations for two sample longitudinal designs assuming known variance. The package also handles the categorical time Mixed Model of Repeated Measures (MMRM) using the formula of Lu, Luo, and Chen (2008)

Details

Package:	longpower
Type:	Package
Version:	1.0
Date:	2013-05-22
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Michael C. Donohue <mdonohue@usc.edu> Anthony C. Gamst Steven D. Edland

References

Diggle PJ, Heagerty PJ, Liang K, Zeger SL. (2002) *Analysis of longitudinal data*. Second Edition. Oxford Statistical Science Series.

Liu, G., & Liang, K. Y. (1997). Sample size calculations for studies with correlated observations. *Biometrics*, 53(3), 937-47.

Lu, K., Luo, X., & Chen, P.-Y. (2008). Sample size estimation for repeated measures analysis in randomized clinical trials with missing data. *International Journal of Biostatistics*, 4, (1)

See Also

`lmmpower`, `power.mmrm`, `power.mmrm.ar1`, `lmmpower`, `diggle.linear.power`, `edland.linear.power`, `liu.liang.linear.power`

`diggle.linear.power` *Sample size calculations for difference in slopes between two groups.*

Description

This function performs the sample size calculation for difference in slopes between two groups. See Diggle, et al (2002) and package vignette for more details.

Usage

```
diggle.linear.power(
  n = NULL,
  delta = NULL,
  t = NULL,
  sigma2 = 1,
  R = NULL,
  sig.level = 0.05,
  power = NULL,
  alternative = c("two.sided", "one.sided"),
  tol = .Machine$double.eps^2
)
```

Arguments

`n`	sample size per group
`delta`	group difference in slopes
`t`	the observation times
`sigma2`	the residual variance
`R`	the working correlation matrix (or variance-covariance matrix if `sigma2` is 1). If `R` is a scalar, an exchangeable working correlation matrix will be assumed.
`sig.level`	Type I error
`power`	power
`alternative`	one- or two-sided test
`tol`	numerical tolerance used in root finding.

Value

The number of subject required per arm to attain the specified `power` given `sig.level` and the other parameter estimates.

Author(s)

Michael C. Donohue, Steven D. Edland

References

Diggle P.J., Heagerty P.J., Liang K., Zeger S.L. (2002) *Analysis of longitudinal data*. Second Edition. Oxford Statistical Science Series.

See Also

`lmmpower`, `diggle.linear.power`

Examples

```
## Not run:
browseVignettes(package = "longpower")

## End(Not run)

# Reproduces the table on page 29 of Diggle et al
n <- 3
t <- c(0,2,5)
rho <- c(0.2, 0.5, 0.8)
sigma2 <- c(100, 200, 300)
tab <- outer(rho, sigma2,
      Vectorize(function(rho, sigma2){
        ceiling(diggle.linear.power(
          delta=0.5,
          t=t,
          sigma2=sigma2,
          R=rho,
          alternative="one.sided",
          power = 0.80)$n[1])}))
colnames(tab) <- paste("sigma2 =", sigma2)
rownames(tab) <- paste("rho =", rho)
tab

# An Alzheimer's Disease example using ADAS-cog pilot estimates
# var of random intercept
sig2.i <- 55
# var of random slope
sig2.s <- 24
# residual var
sig2.e <- 10
# covariance of slope and intercep
cov.s.i <- 0.8*sqrt(sig2.i)*sqrt(sig2.s)
```

```
cov.t <- function(t1, t2, sig2.i, sig2.s, cov.s.i){
        sig2.i + t1*t2*sig2.s + (t1+t2)*cov.s.i
}

t <- seq(0,1.5,0.25)
n <- length(t)
R <- outer(t, t, function(x,y){cov.t(x,y, sig2.i, sig2.s, cov.s.i)})
R <- R + diag(sig2.e, n, n)

diggle.linear.power(d=1.5, t=t, R=R, sig.level=0.05, power=0.80)
```

`edland.linear.power` *Linear mixed model sample size calculations.*

Description

This function performs sample size calculations for the linear mixed model with random intercepts and slopes when used to test for differences in fixed effects slope between groups. Input parameters are random effect variance and residual error variance as estimated by a REML fit to representative pilot data or data from a representative prior clinical trial or cohort study.

Usage

```
edland.linear.power(
  n = NULL,
  delta = NULL,
  power = NULL,
  t = NULL,
  lambda = 1,
  sig2.int = 0,
  sig2.s = NULL,
  sig.b0b1 = 0,
  sig2.e = NULL,
  sig2.int_2 = NULL,
  sig2.s_2 = NULL,
  sig.b0b1_2 = NULL,
  sig2.e_2 = NULL,
  sig.level = 0.05,
  p = NULL,
  p_2 = NULL,
  alternative = c("two.sided", "one.sided"),
  tol = NULL
)
```

Arguments

n	sample size, group 1
delta	group difference in fixed effect slopes
power	power
t	the observation times
lambda	allocation ratio (= (sample size group 1)/(sample size group 2)
sig2.int	variance of random intercepts, group 1
sig2.s	variance of random slopes, group 1
sig.b0b1	covariance of random slopes and intercepts,group 1
sig2.e	residual variance, group 1
sig2.int_2	variance of random intercepts, group 2 (defaults to sig2.int)
sig2.s_2	variance of random slopes, group 2 (defaults to sig2.s)
sig.b0b1_2	covariance of random slopes and intercepts, group 2 (defaults to sig.b0b1)
sig2.e_2	residual variance, group 2 (defaults to sig2.e)
sig.level	type one error
p	proportion vector for group 1, if i indexes visits, p[i] = the proportion whose last visit was at visit i (p sums to 1)
p_2	proportion vector for group 2 (defaults to p)
alternative	one- or two-sided test
tol	not used (no root finding used in this implementation).

Details

Default settings perform sample size / power / effect size calculations assuming equal covariance of repeated measures in the 2 groups, equal residual error variance across groups, equal allocation to groups, and assuming no study subject attrition. Specifically, variance parameters required for default settings are sig2.s, the variance of random slopes, and sig2.e, the residual error variance, both either known or estimated from a mixed model fit by REML to prior data.

This function will also provide sample size estimates for linear mixed models with random intercept only by setting sig2.s = 0 (although, this is not generally recommended).

This function was generalized April 2020. The function is back compatible, although the order of arguments has changed. The new function accommodates different variance parameters across groups, unequal allocation across groups, and study subject attrition (loss to followup), which may also vary across groups.

- Unequal allocation is accommodated by the parameter lambda, where lambda = (sample size group 1)/(sample size group 2). lambda defaults to one (equal allocation).
- Study subject attrition is accommodated by the parameter 'p', where p is a vector of proportions. If i indexes successive study visits, p[i] = the proportion whose last visit is at visit i. p sums to 1. p defaults to the case of no study subject attrition (everyone completes all visits).
- differential study subject attrition is accommodated by the parameter p_2. p_2 is analogous to p, but for group 2. p_2 defaults to p (equal pattern of study subject attrition across groups).

- Note that when there is study subject attrition, sample size / power calculations are also a function of the variance of random intercepts and the covariance of random intercepts and slopes. When p and/or p_2 are specified, edland.linear.power requires specification of these parameters. (These are part of the standard output of lmer and other software fitting REML models.) These parameters are specified by sig2.int and sig.b0b1 (group 1), and sig2.int_2 and sigb0b1_2 (group 2).
- different variance parameters across groups is accommodated by the variance arguments sig2.int_2, sig.b0b1_2, sig2.s_2 and sig2.e_2, analogous to the the corresponding arguments within group 1. These values default to to the corresponding group 1 variables (equal variance across groups).
- The parameter t is the design vector. For example, a one year trial with observations every three months would specify t = c(0,.25,.5,.75,1).

Value

One of the number of subject required per arm, the power, or detectible effect size given sig.level and the other parameter estimates.

Author(s)

Michael C. Donohue, Steven D. Edland

References

Ard and Edland, S.D. (2011) Power calculations for clinical trials in Alzheimer's disease. *Journal of Alzheimer's Disease.* 21:369-377.

See Also

lmmpower, diggle.linear.power, liu.liang.linear.power

Examples

```
## Not run:
browseVignettes(package = "longpower")

## End(Not run)
# An Alzheimer's Disease example using ADAS-cog pilot estimates
t <- seq(0,1.5,0.25)
edland.linear.power(delta=1.5, t=t, sig2.s = 24, sig2.e = 10, sig.level=0.05, power = 0.80)
```

`liu.liang.linear.power` *Linear mixed model sample size calculations from Liu & Liang (1997).*

Description

This function performs the sample size calculation for a linear mixed model. See Liu and Liang (1997) for parameter definitions and other details.

Usage

```
liu.liang.linear.power(
  N = NULL,
  delta = NULL,
  u = NULL,
  v = NULL,
  sigma2 = 1,
  R = NULL,
  R.list = NULL,
  sig.level = 0.05,
  power = NULL,
  Pi = rep(1/length(u), length(u)),
  alternative = c("two.sided", "one.sided"),
  tol = .Machine$double.eps^2
)
```

Arguments

`N`	The total sample size. This formula can accommodate unbalanced group allocation via `Pi`. See Liu and Liang (1997) for more details
`delta`	group difference (possibly a vector of differences)
`u`	a list of covariate vectors or matrices associated with the parameter of interest
`v`	a respective list of covariate vectors or matrices associated with the nuisance parameter
`sigma2`	the error variance
`R`	the variance-covariance matrix for the repeated measures
`R.list`	a list of variance-covariance matrices for the repeated measures, if assumed different in two groups
`sig.level`	type one error
`power`	power
`Pi`	the proportion of covariates of each type
`alternative`	one- or two-sided test
`tol`	numerical tolerance used in root finding.

Details

The parameters u, v, and Pi are expected to be the same length and sorted with respect to each other. See Liu and Liang (1997) and package vignette for more details.

References

Liu, G. and Liang, K. Y. (1997) Sample size calculations for studies with correlated observations. *Biometrics*, 53(3), 937-47.

See Also

lmmpower

Examples

```
## Not run:
browseVignettes(package = "longpower")

## End(Not run)

# Reproduces the table on page 29 of Diggle et al for
# difference in slopes between groups

n <- 3
t <- c(0,2,5)
u <- list(u1 = t, u2 = rep(0,n))
v <- list(v1 = cbind(1,1,t),
         v2 = cbind(1,0,t))
rho <- c(0.2, 0.5, 0.8)
sigma2 <- c(100, 200, 300)
tab <- outer(rho, sigma2,
      Vectorize(function(rho, sigma2){
        ceiling(liu.liang.linear.power(
          delta=0.5, u=u, v=v,
          sigma2=sigma2,
          R=rho, alternative="one.sided",
          power=0.80)$N/2)}))
colnames(tab) <- paste("sigma2 =", sigma2)
rownames(tab) <- paste("rho =", rho)
tab

# Reproduces the table on page 30 of Diggle et al for
# difference in average response between groups.

n <- 3
u <- list(u1 = rep(1,n), u2 = rep(0,n))
v <- list(v1 = rep(1,n),
         v2 = rep(1,n))
rho <- c(0.2, 0.5, 0.8)
delta <- c(20, 30, 40, 50)/100
tab <- outer(rho, delta,
```

```
     Vectorize(function(rho, delta){
       ceiling(liu.liang.linear.power(
         delta=delta, u=u, v=v,
         sigma2=1,
         R=rho, alternative="one.sided",
         power=0.80)$n[1])}))
colnames(tab) <- paste("delta =", delta)
rownames(tab) <- paste("rho =", rho)
tab

# An Alzheimer's Disease example using ADAS-cog pilot estimates
# var of random intercept
sig2.i <- 55
# var of random slope
sig2.s <- 24
# residual var
sig2.e <- 10
# covariance of slope and intercep
cov.s.i <- 0.8*sqrt(sig2.i)*sqrt(sig2.s)

cov.t <- function(t1, t2, sig2.i, sig2.s, cov.s.i){
        sig2.i + t1*t2*sig2.s + (t1+t2)*cov.s.i
}

t <- seq(0,1.5,0.25)
n <- length(t)
R <- outer(t, t, function(x,y){cov.t(x,y, sig2.i, sig2.s, cov.s.i)})
R <- R + diag(sig2.e, n, n)
u <- list(u1 = t, u2 = rep(0,n))
v <- list(v1 = cbind(1,1,t),
         v2 = cbind(1,0,t))

liu.liang.linear.power(delta=1.5, u=u, v=v, R=R, sig.level=0.05, power=0.80)
liu.liang.linear.power(N=416, u=u, v=v, R=R, sig.level=0.05, power=0.80)
liu.liang.linear.power(N=416, delta = 1.5, u=u, v=v, R=R, sig.level=0.05)
liu.liang.linear.power(N=416, delta = 1.5, u=u, v=v, R=R, power=0.80, sig.level = NULL)

# Reproduces total sample sizes, m, of Table 1 of Liu and Liang 1997
tab1 <- data.frame(cbind(
  n = c(rep(4, 4), rep(2, 4), 1),
  rho = c(0.0, 0.3, 0.5, 0.8)))
m <- c()
for(i in 1:nrow(tab1)){
  R <- matrix(tab1$rho[i], nrow = tab1$n[i], ncol = tab1$n[i])
  diag(R) <- 1
  m <- c(m, ceiling(liu.liang.linear.power(
    delta=0.5,
    u = list(u1 = rep(1, tab1$n[i]), # treatment
             u2 = rep(0, tab1$n[i])), # control
    v = list(v1 = rep(1, tab1$n[i]), v2 = rep(1, tab1$n[i])), # intercept
    sigma2=1,
    R=R, alternative="two.sided",
    power=0.90)$N))
```

```
}
cbind(tab1, m)

# Reproduces total sample sizes, m, of Table 3.a. of Liu and Liang 1997
# with unbalanced design
tab3 <- data.frame(cbind(
  rho = rep(c(0.0, 0.3, 0.5, 0.8), 2),
  pi1 = c(rep(0.8, 4), rep(0.2, 4))))
m <- c()
for(i in 1:nrow(tab3)){
  R <- matrix(tab3$rho[i], nrow = 4, ncol = 4)
  diag(R) <- 1
  m <- c(m, ceiling(liu.liang.linear.power(
    delta=0.5,
    u = list(u1 = rep(1, 4), # treatment
             u2 = rep(0, 4)), # control
    v = list(v1 = rep(1, 4), v2 = rep(1, 4)), # intercept
    sigma2=1,
    Pi = c(tab3$pi1[i], 1-tab3$pi1[i]),
    R=R, alternative="two.sided",
    power=0.90)$N))
}
cbind(tab3, m)
```

`lmmpower` *Sample size calculations for linear mixed models of rate of change based on lmer, lme, or gee "placebo" pilot estimates.*

Description

These functions compute sample size for linear mixed models based on the formula due to Diggle (2002) or Liu and Liang (1997). These formulae are expressed in terms of marginal model or Generalized Estimating Equations (GEE) parameters. These functions translate pilot mixed effect model parameters (e.g. random intercept and/or slope, fixed effects, etc.) into marginal model parameters so that either formula can be applied to equivalent affect. Pilot estimates are assumed to be from an appropriate "placebo" group and the parameter of interest is assumed to be the rate of change over time of the outcome.

Usage

```
## Default S3 method:
lmmpower(
  object = NULL,
  n = NULL,
  parameter = 2,
  pct.change = NULL,
  delta = NULL,
  t = NULL,
```

```
  sig.level = 0.05,
  power = NULL,
  alternative = c("two.sided", "one.sided"),
  beta = NULL,
  beta.CI = NULL,
  delta.CI = NULL,
  sig2.i = NULL,
  sig2.s = NULL,
  sig2.e = NULL,
  cov.s.i = NULL,
  R = NULL,
  method = c("diggle", "liuliang", "edland"),
  tol = .Machine$double.eps^2,
  ...
)
```

Arguments

object	an object returned by lme4
n	sample size per group of a mixed-effects model object to placebo data assumed to have either a random intercept, or a random intercept and random effect for time (slope); and fixed effect representing the rate of change in a placebo group.
parameter	the name or position of the rate of change parameter of interest, e.g. ("time", "t", or 2 if it is the second specified fixed effect).
pct.change	the percent change in the pilot estimate of the parameter of interest (beta, the placebo/null effect)
delta	the change in the pilot estimate of the parameter of interest, computed from pct.change if left missing.
t	vector of time points
sig.level	Type I error
power	power
alternative	"two.sided" or "one.sided"
beta	pilot estimate of the placebo effect (slope or rate of change in the outcome)
beta.CI	95% confidence limits of the pilot estimate of beta
delta.CI	95% confidence limits of the effect size
sig2.i	pilot estimate of variance of random intercept
sig2.s	pilot estimate of variance of random slope
sig2.e	pilot estimate of residual variance
cov.s.i	pilot estimate of covariance of random slope and intercept
R	pilot estimate of a marginal model working correlation matrix
method	the formula to use. Defaults to "diggle" for Diggle et al (2002). Alternatively "liuliang" can be selected for Liu & Liang (1997), or "edland" for Ard & Edland (2011).
tol	numerical tolerance used in root finding.
...	other arguments

Details

Any parameters not explicitly stated are extracted from the fitted `object`.

Value

An object of class `power.htest` giving the calculated sample size, N, per group and other parameters.

Author(s)

Michael C. Donohue

References

Diggle P.J., Heagerty P.J., Liang K., Zeger S.L. (2002) *Analysis of longitudinal data*. Second Edition. Oxford Statistical Science Series.

Liu, G., and Liang, K. Y. (1997) Sample size calculations for studies with correlated observations. *Biometrics*, 53(3), 937-47.

Ard, C. and Edland, S.D. (2011) Power calculations for clinical trials in Alzheimer's disease. *Journal of Alzheimer's Disease.* 21:369-377.

See Also

`liu.liang.linear.power`, `diggle.linear.power`, `edland.linear.power`

Examples

```
## Not run:
browseVignettes(package = "longpower")

## End(Not run)

lmmpower(delta=1.5, t = seq(0,1.5,0.25),
sig2.i = 55, sig2.s = 24, sig2.e = 10, cov.s.i=0.8*sqrt(55)*sqrt(24), power = 0.80)
lmmpower(n=208, t = seq(0,1.5,0.25),
sig2.i = 55, sig2.s = 24, sig2.e = 10, cov.s.i=0.8*sqrt(55)*sqrt(24), power = 0.80)
lmmpower(beta = 5, pct.change = 0.30, t = seq(0,1.5,0.25),
sig2.i = 55, sig2.s = 24, sig2.e = 10, cov.s.i=0.8*sqrt(55)*sqrt(24), power = 0.80)

## Not run:
library(lme4)
fm1 <- lmer(Reaction ~ Days + (Days|Subject), sleepstudy)
lmmpower(fm1, pct.change = 0.30, t = seq(0,9,1), power = 0.80)

library(nlme)
fm2 <- lme(Reaction ~ Days, random=~Days|Subject, sleepstudy)
lmmpower(fm2, pct.change = 0.30, t = seq(0,9,1), power = 0.80)

# random intercept only
```

```
fm3 <- lme(Reaction ~ Days, random=~1|Subject, sleepstudy)
lmmpower(fm3, pct.change = 0.30, t = seq(0,9,1), power = 0.80)

library(gee)
fm4 <- gee(Reaction ~ Days, id = Subject,
           data = sleepstudy,
           corstr = "exchangeable")
lmmpower(fm4, pct.change = 0.30, t = seq(0,9,1), power = 0.80)

## End(Not run)
```

`power.longtest` *Constructor function for class* `"power.longtest"`

Description

Constructor function for class `"power.longtest"`

Usage

```
power.longtest(object)
```

Arguments

`object`	a list.

Value

an object of class `"power.longtest"`

`power.mmrm` *Linear mixed model sample size calculations.*

Description

This function performs the sample size calculation for a mixed model of repeated measures with general correlation structure. See Lu, Luo, & Chen (2008) for parameter definitions and other details. This function executes Formula (3) on page 4.

Usage

```
power.mmrm(
  N = NULL,
  Ra = NULL,
  ra = NULL,
  sigmaa = NULL,
  Rb = NULL,
  rb = NULL,
  sigmab = NULL,
  lambda = 1,
  delta = NULL,
  sig.level = 0.05,
  power = NULL,
  alternative = c("two.sided", "one.sided"),
  tol = .Machine$double.eps^2
)
```

Arguments

N	total sample size
Ra	correlation matrix for group a
ra	retention in group a
sigmaa	standard deviation of observation of interest in group a
Rb	correlation matrix for group a
rb	retention in group b
sigmab	standard deviation of observation of interest in group b. If NULL, sigmab is assumed same as sigmaa. If not NULL, sigmaa and sigmab are averaged.
lambda	allocation ratio
delta	effect size
sig.level	type one error
power	power
alternative	one- or two-sided test
tol	numerical tolerance used in root finding.

Details

See Lu, Luo, & Chen (2008).

Value

The number of subject required per arm to attain the specified power given sig.level and the other parameter estimates.

Author(s)

Michael C. Donohue

References

Lu, K., Luo, X., Chen, P.-Y. (2008) Sample size estimation for repeated measures analysis in randomized clinical trials with missing data. *International Journal of Biostatistics*, 4, (1)

See Also

power.mmrm.ar1, lmmpower, diggle.linear.power

Examples

```
# reproduce Table 1 from Lu, Luo, & Chen (2008)
phi1 <- c(rep(1, 6), 2, 2)
phi2 <- c(1, 1, rep(2, 6))
lambda <- c(1, 2, sqrt(1/2), 1/2, 1, 2, 1, 2)
ztest <- ttest1 <- c()
for(i in 1:8){
  Na <- (phi1[i] + lambda[i] * phi2[i])*(qnorm(0.05/2) + qnorm(1-0.90))^2*(0.5^-2)
  Nb <- Na/lambda[i]
  ztest <- c(ztest, Na + Nb)
  v <- Na + Nb - 2
 Na <- (phi1[i] + lambda[i] * phi2[i])*(qt(0.05/2, df = v) + qt(1-0.90, df = v))^2*(0.5^-2)
  Nb <- Na/lambda[i]
  ttest1 <- c(ttest1, Na + Nb)
}
data.frame(phi1, phi2, lambda, ztest, ttest1)

Ra <- matrix(0.25, nrow = 4, ncol = 4)
diag(Ra) <- 1
ra <- c(1, 0.90, 0.80, 0.70)
sigmaa <- 1
power.mmrm(Ra = Ra, ra = ra, sigmaa = sigmaa, delta = 0.5, power = 0.80)
power.mmrm(N = 174, Ra = Ra, ra = ra, sigmaa = sigmaa, delta = 0.5)
power.mmrm(N = 174, Ra = Ra, ra = ra, sigmaa = sigmaa, power = 0.80)

power.mmrm(Ra = Ra, ra = ra, sigmaa = sigmaa, delta = 0.5, power = 0.80, lambda = 2)
power.mmrm(N = 174, Ra = Ra, ra = ra, sigmaa = sigmaa, delta = 0.5, lambda = 2)
power.mmrm(N = 174, Ra = Ra, ra = ra, sigmaa = sigmaa, power = 0.80, lambda = 2)

# Extracting paramaters from gls objects with general correlation

# Create time index:
Orthodont$t.index <- as.numeric(factor(Orthodont$age, levels = c(8, 10, 12, 14)))
with(Orthodont, table(t.index, age))

fmOrth.corSym <- gls( distance ~ Sex * I(age - 11),
  Orthodont,
  correlation = corSymm(form = ~ t.index | Subject),
  weights = varIdent(form = ~ 1 | age) )
summary(fmOrth.corSym)$tTable

C <- corMatrix(fmOrth.corSym$modelStruct$corStruct)[[1]]
```

```
sigmaa <- fmOrth.corSym$sigma *
          coef(fmOrth.corSym$modelStruct$varStruct, unconstrained = FALSE)['14']
ra <- seq(1,0.80,length=nrow(C))
power.mmrm(N=100, Ra = C, ra = ra, sigmaa = sigmaa, power = 0.80)

# Extracting paramaters from gls objects with compound symmetric correlation

fmOrth.corCompSymm <- gls( distance ~ Sex * I(age - 11),
  Orthodont,
  correlation = corCompSymm(form = ~ t.index | Subject),
  weights = varIdent(form = ~ 1 | age) )
summary(fmOrth.corCompSymm)$tTable

C <- corMatrix(fmOrth.corCompSymm$modelStruct$corStruct)[[1]]
sigmaa <- fmOrth.corCompSymm$sigma *
          coef(fmOrth.corCompSymm$modelStruct$varStruct, unconstrained = FALSE)['14']
ra <- seq(1,0.80,length=nrow(C))
power.mmrm(N=100, Ra = C, ra = ra, sigmaa = sigmaa, power = 0.80)

# Extracting paramaters from gls objects with AR1 correlation

fmOrth.corAR1 <- gls( distance ~ Sex * I(age - 11),
  Orthodont,
  correlation = corAR1(form = ~ t.index | Subject),
  weights = varIdent(form = ~ 1 | age) )
summary(fmOrth.corAR1)$tTable

C <- corMatrix(fmOrth.corAR1$modelStruct$corStruct)[[1]]
sigmaa <- fmOrth.corAR1$sigma *
          coef(fmOrth.corAR1$modelStruct$varStruct, unconstrained = FALSE)['14']
ra <- seq(1,0.80,length=nrow(C))
power.mmrm(N=100, Ra = C, ra = ra, sigmaa = sigmaa, power = 0.80)
power.mmrm.ar1(N=100, rho = C[1,2], ra = ra, sigmaa = sigmaa, power = 0.80)
```

---

power.mmrm.ar1 *Linear mixed model sample size calculations.*

Description

This function performs the sample size calculation for a mixed model of repeated measures with AR(1) correlation structure. See Lu, Luo, & Chen (2008) for parameter definitions and other details.

Usage

```
power.mmrm.ar1(
  N = NULL,
  rho = NULL,
  ra = NULL,
  sigmaa = NULL,
```

```
  rb = NULL,
  sigmab = NULL,
  lambda = 1,
  times = 1:length(ra),
  delta = NULL,
  sig.level = 0.05,
  power = NULL,
  alternative = c("two.sided", "one.sided"),
  tol = .Machine$double.eps^2
)
```

Arguments

N	total sample size
rho	AR(1) correlation parameter
ra	retention in group a
sigmaa	standard deviation of observation of interest in group a
rb	retention in group a (assumed same as ra if left blank)
sigmab	standard deviation of observation of interest in group b. If NULL, sigmab is assumed same as sigmaa. If not NULL, sigmaa and sigmab are averaged.
lambda	allocation ratio
times	observation times
delta	effect size
sig.level	type one error
power	power
alternative	one- or two-sided test
tol	numerical tolerance used in root finding.

Details

See Lu, Luo, & Chen (2008).

Value

The number of subject required per arm to attain the specified power given sig.level and the other parameter estimates.

Author(s)

Michael C. Donohue

References

Lu, K., Luo, X., Chen, P.-Y. (2008) Sample size estimation for repeated measures analysis in randomized clinical trials with missing data. *International Journal of Biostatistics*, 4, (1)

See Also

power.mmrm, lmmpower, diggle.linear.power

Examples

```
# reproduce Table 2 from Lu, Luo, & Chen (2008)
tab <- c()
for(J in c(2,4))
for(aJ in (1:4)/10)
for(p1J in c(0, c(1, 3, 5, 7, 9)/10)){
  rJ <- 1-aJ
  r <- seq(1, rJ, length = J)
  # p1J = p^(J-1)
  tab <- c(tab, power.mmrm.ar1(rho = p1J^(1/(J-1)), ra = r, sigmaa = 1,
    lambda = 1, times = 1:J,
    delta = 1, sig.level = 0.05, power = 0.80)$phi1)
}
matrix(tab, ncol = 6, byrow = TRUE)

# approximate simulation results from Table 5 from Lu, Luo, & Chen (2008)
ra <- c(100, 76, 63, 52)/100
rb <- c(100, 87, 81, 78)/100

power.mmrm.ar1(rho=0.6, ra=ra, sigmaa=1, rb = rb,
               lambda = sqrt(1.25/1.75), power = 0.904, delta = 0.9)
power.mmrm.ar1(rho=0.6, ra=ra, sigmaa=1, rb = rb,
               lambda = 1.25/1.75, power = 0.910, delta = 0.9)
power.mmrm.ar1(rho=0.6, ra=ra, sigmaa=1, rb = rb,
               lambda = 1, power = 0.903, delta = 0.9)
power.mmrm.ar1(rho=0.6, ra=ra, sigmaa=1, rb = rb,
               lambda = 2, power = 0.904, delta = 0.9)

power.mmrm.ar1(N=81, ra=ra, sigmaa=1, rb = rb,
               lambda = sqrt(1.25/1.75), power = 0.904, delta = 0.9)
power.mmrm.ar1(N=87, rho=0.6, ra=ra, sigmaa=1, rb = rb,
               lambda = 1.25/1.75, power = 0.910)
power.mmrm.ar1(N=80, rho=0.6, ra=ra, sigmaa=1, rb = rb,
               lambda = 1, delta = 0.9)
power.mmrm.ar1(N=84, rho=0.6, ra=ra, sigmaa=1, rb = rb,
               lambda = 2, power = 0.904, delta = 0.9, sig.level = NULL)

# Extracting paramaters from gls objects with AR1 correlation

# Create time index:
Orthodont$t.index <- as.numeric(factor(Orthodont$age, levels = c(8, 10, 12, 14)))
with(Orthodont, table(t.index, age))

fmOrth.corAR1 <- gls( distance ~ Sex * I(age - 11),
  Orthodont,
  correlation = corAR1(form = ~ t.index | Subject),
  weights = varIdent(form = ~ 1 | age) )
```

```
summary(fmOrth.corAR1)$tTable

C <- corMatrix(fmOrth.corAR1$modelStruct$corStruct)[[1]]
sigmaa <- fmOrth.corAR1$sigma *
          coef(fmOrth.corAR1$modelStruct$varStruct, unconstrained = FALSE)['14']
ra <- seq(1,0.80,length=nrow(C))
power.mmrm(N=100, Ra = C, ra = ra, sigmaa = sigmaa, power = 0.80)
power.mmrm.ar1(N=100, rho = C[1,2], ra = ra, sigmaa = sigmaa, power = 0.80)
```

`print.power.longtest` *Print method for longitudinal data power calculation object*

Description

Print object of class `"power.longtest"` in nice layout.

Usage

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

Arguments

`x`	Object of class `"power.longtest"`.
`...`	further arguments to be passed to or from methods.

Details

A `power.longtest` object is just a named list of numbers and character strings, supplemented with `method` and `note` elements. The `method` is displayed as a title, the `note` as a footnote, and the remaining elements are given in an aligned 'name = value' format.

Value

none

See Also

`liu.liang.linear.power`, `diggle.linear.power`, `lmmpower`,

Index