Package ‘longpower’

July 25, 2016

**Type** Package

**Title** Sample Size Calculations for Longitudinal Data

**Version** 1.0-16

**Date** 2016-07-24

**Author** Michael C. Donohue, Anthony C. Gamst, Steven D. Edland

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

**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 (2002). 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.

**License** GPL (>= 2)

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

**Suggests** knitr, gee, testthat, methods

**LazyLoad** yes

**VignetteBuilder** knitr

**URL** https://bitbucket.org/mdonohue/longpower

**Collate** 'longpower-package.R' 'diggle.linear.power.R'

'edland.linear.power.R' 'liu.liang.linear.power.R' 'lmm.power.R'

'power_mmrm.R' 'print.power.longtest.R'

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-07-25 20:36:27
R topics documented:

longpower-package .................................................. 2
diggle.linear.power ................................................ 3
edland.linear.power ............................................... 5
liu.liang.linear.power ............................................. 6
lmmpower ............................................................... 10
power.longtest ...................................................... 12
power.mmrm ........................................................... 13
power.mmrm.ar1 ..................................................... 15
print.power.longtest ................................................ 18

Index 19

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


**See Also**

`lmmpower, power.mmmr, power.mmmr.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) for parameter definitions and other details.

**Usage**

```r
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.

**Details**


**Value**

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

Author(s)
Michael C. Donohue, Steven D. Edland

References

See Also
lmm.power, diggle.linear.power

Examples

```r
## 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))}))
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 intercept
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
}
Description

This function performs the sample size calculation for a linear mixed model with random slope.

Usage

```r
edland.linear.power(n = NULL, delta = NULL, t = NULL, sig2.s = 0,
                    sig2.e = 1, 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
- `sig2.s`: variance of random slope
- `sig2.e`: residual variance
- `sig.level`: type one error
- `power`: power
- `alternative`: one- or two-sided test
- `tol`: numerical tolerance used in root finding.

Details

This function will also provide sample size estimates for linear mixed models with random intercept only simply by setting `sig2.s = 0`.

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


### See Also

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

### Examples

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

## End(Not run)
# Reproduces the table on page 29 of Diggle et al
n = 3
t = c(0.2, 0.5)
rho = c(0.2, 0.5, 0.8)
sigma2 = c(100, 200, 300)
tab = outer(rho, sigma2,
  Vectorize(function(rho, sigma2){
    ceiling(edland.linear.power(
      delta=0.5,
      t=t,
      sig2.e=sigma2*(1-rho),
      alternative="one.sided",
      power=0.80)$n))}))
colnames(tab) = paste("sigmaR =", sigma2)
rownames(tab) = paste("rho =", rho)
tab

# An Alzheimer's Disease example using ADAS-cog pilot estimates
t = seq(0, 1.5, 0.25)
n = length(t)
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


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("sigma =", 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/2))
    )
    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 intercept
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
liu.liang.linear.power

```r

# Reproduces total sample sizes, m, of Table 1 of Liu and Liang 1997

t = seq(0, 1.5, 0.25)
n = length(t)
R = outer(t, t, function(x, y){cov(t(x, y), sig2.1, 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, t),
v2 = cbind(1, 0, t))

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

# Reproduces total sample sizes, m, of Table 3.a. of Liu and Liang 1997
# with unbalanced design

for(i in 1:nrow(tab1)){
R <- matrix(tab1$rho[i], nrow = tab1$n[i], ncol = tab1$n[i])
diag(R) <- 1
m <- ceiling(liu.liang.linear.power(
    delta=0.5,
    u = list(u1 = rep(1, tab1$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)
}

```
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 effect. 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

lmmpower.default(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("edland", "diggle", "liuliang"), 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
lmmpower

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
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.h.test giving the calculated sample size, N, per group and other parameters.

Author(s)
Michael C. Donohue

References

See Also
liu.liang.linear.power diggle.linear.power

Examples

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

lmmpower(delta=1.5, t = seq(0,1.5,0.25),
sig2.1 = 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),
sig2.1 = 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),
```


sig2.1 = 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.mmmr**

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**

```r
power.mmmr(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**


**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


See Also

`power.mmmr.marl, lmmpower, diggle.linear.power`

Examples

```r
# 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 <- ttest <- c()
for(i in 1:8){
  Na <- (phi1[i] + lambda[i] * phi2[i]) * (qnorm(0.05/2) + qnorm(1-0.99))^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(1-0.99, df = v))^2 * 0.5^2
  Nb <- Na/lambda[i]
  ttest <- c(ttest, Na + Nb)
}
data.frame(phi1, phi2, lambda, ztest, ttest)

Ra <- matrix(0.25, nrow = 4, ncol = 4)
diag(Ra) <- 1
ra <- c(1, 0.90, 0.80, 0.70)
sigmaa <- 1
power.mmmr(Ra = Ra, ra = ra, sigmaa = sigmaa, delta = 0.5, power = 0.80)
power.mmmr(N = 174, Ra = Ra, ra = ra, sigmaa = sigmaa, delta = 0.5)
power.mmmr(N = 174, Ra = Ra, ra = ra, sigmaa = sigmaa, power = 0.80)
power.mmmr(N = 174, Ra = Ra, ra = ra, sigmaa = sigmaa, delta = 0.5, lambda = 2)
power.mmmr(N = 174, Ra = Ra, ra = ra, sigmaa = sigmaa, delta = 0.5, lambda = 2)
power.mmmr(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,]
```
`power.mmmr.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

```r
power.mmmr.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


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


See Also

`power.mmmrm, lmmpower, diggle.linear.power`

Examples

```r
# reproduce Table 2 from Lu, Luo, & Chen (2008)
tab <- c()
for(J in c(2,4))
  for(aj in (1:4)/10)
    for(piJ in c(0, c(1, 3, 5, 7, 9)/10)){
```
> rJ <- 1:J
> r <- seq(1, rJ, length = J)
> # p|J = p^{J-1}
> tab <- c(tab, power.mmmr.ar1(rho = pJ|1/((J-1)), ra = r, sigmaa = 1,
>                             lambda = 1, times = 1:J,
>                             delta = 1, sig.level = 0.05, power = 0.80))
>
> 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.mmmr.ar1(rho=0.6, ra=ra, sigmaa=1, rb = rb,
>                lambda = sqrt(1.25/1.75), power = 0.904, delta = 0.9)
> power.mmmr.ar1(rho=0.6, ra=ra, sigmaa=1, rb = rb,
>                lambda = 1.25/1.75, power = 0.910, delta = 0.9)
> power.mmmr.ar1(rho=0.6, ra=ra, sigmaa=1, rb = rb,
>                lambda = 1, power = 0.903, delta = 0.9)
> power.mmmr.ar1(rho=0.6, ra=ra, sigmaa=1, rb = rb,
>                lambda = 2, power = 0.904, delta = 0.9)

> power.mmmr.ar1(N=81, ra=ra, sigmaa=1, rb = rb,
>                lambda = sqrt(1.25/1.75), power = 0.904, delta = 0.9)
> power.mmmr.ar1(N=87, rho=0.6, ra=ra, sigmaa=1, rb = rb,
>                lambda = 1.25/1.75, power = 0.910)
> power.mmmr.ar1(N=88, rho=0.6, ra=ra, sigmaa=1, rb = rb,
>                lambda = 1, delta = 0.9)
> power.mmmr.ar1(N=84, rho=0.6, ra=ra, sigmaa=1, rb = rb,
>                lambda = 2, power = 0.904, delta = 0.9, sig.level = NULL)

# Extracting parameters 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 <- coef(fmOrth.corAR1$modelStruct$varStruct, unconstrained = FALSE)[14]
ra <- seq(1,0.80,length=nrow(C))

> power.mmmr(N=100, Ra = C, r = ra, sigmaa = sigmaa, power = 0.80)
> power.mmmr.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
```r
## 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`, `lmm.power`,
Index

*Topic **effects**
  diggle.linear.power, 3
  edland.linear.power, 5
  liu.liang.linear.power, 6
  lmm.power, 10
  power.mmmrm, 13
  power.mmmrm.ar1, 15

*Topic **longtest**
  print.power.longtest, 18

*Topic **marginal**
  lmm.power, 10

*Topic **methods**
  lmm.power, 10

*Topic **mixed**
  diggle.linear.power, 3
  edland.linear.power, 5
  liu.liang.linear.power, 6
  lmm.power, 10
  power.mmmrm, 13
  power.mmmrm.ar1, 15

*Topic **model**
  lmm.power, 10

*Topic **package**
  longpower-package, 2

*Topic **power**
  diggle.linear.power, 3
  edland.linear.power, 5
  liu.liang.linear.power, 6
  lmm.power, 10
  power.mmmrm, 13
  power.mmmrm.ar1, 15

*Topic **random**
  diggle.linear.power, 3
  edland.linear.power, 5
  liu.liang.linear.power, 6
  lmm.power, 10
  power.mmmrm, 13
  power.mmmrm.ar1, 15

*Topic **sample**
  diggle.linear.power, 3
  edland.linear.power, 5
  liu.liang.linear.power, 6
  lmm.power, 10
  power.mmmrm, 13
  power.mmmrm.ar1, 15

*Topic **size**
  diggle.linear.power, 3
  edland.linear.power, 5
  liu.liang.linear.power, 6
  lmm.power, 10
  power.mmmrm, 13
  power.mmmrm.ar1, 15

  diggle.linear.power, 3, 3, 4, 6, 11, 14, 16, 18
  edland.linear.power, 3, 5
  liu.liang.linear.power, 3, 6, 6, 11, 18
  lmm.power, 3, 4, 6, 7, 10, 14, 16, 18
  lmm.power, ANY-method (lmm.power), 10
  lmm.power, merMod-method (lmm.power), 10
  lmm.power-methods (lmm.power), 10
  lmm.power.default (lmm.power), 10
  lmm.power.gee (lmm.power), 10
  lmm.power.lme (lmm.power), 10
  lmm.power.numeric (lmm.power), 10
  longpower-package, 2

  power.longtest, 12
  power.mmmrm, 3, 13, 16
  power.mmmrm.ar1, 3, 14, 15
  print.power.longtest, 18