Package ‘pamm’

July 23, 2025

Type Package

Title Power Analysis for Random Effects in Mixed Models

Version 1.122

Date 2023-08-28

URL `https://github.com/JulienGAMartin/pamm_R`

BugReports `https://github.com/JulienGAMartin/pamm_R/issues`

Description Simulation functions to assess or explore the power of a dataset to estimates significant random effects (intercept or slope) in a mixed model. The functions are based on the ``lme4'' and ``lmerTest'' packages.

License GPL-2 | GPL-3

Depends R (>= 3.0.0)

Imports lmerTest, lattice, mvtnorm, lme4

Suggests rgl

LazyLoad yes

RoxygenNote 7.2.3

NeedsCompilation no

Author Julien Martin [aut, cre]

Maintainer Julien Martin <`julien.martin@uottawa.ca`>

Repository CRAN

Date/Publication 2023-08-28 21:40:02 UTC

Contents

`pamm-package` *pamm: Power Analysis for Random Effects in Mixed Models*

Description

Simulation functions to assess or explore the power of a dataset to estimates significant random effects (intercept or slope) in a mixed model. The functions are based on the "lme4" and "lmerTest" packages.

Author(s)

Maintainer: Julien Martin `<julien.martin@uottawa.ca>`

See Also

Useful links:

- `https://github.com/JulienGAMartin/pamm_R`
- Report bugs at `https://github.com/JulienGAMartin/pamm_R/issues`

`EAMM` *Simulation function for exploratory power analysis for random effects*

Description

Given a specific sample size, fixed number of group and replicates per group, the function simulate different variance-covariance structure and assess p-values and power of random intercept and random slope

Usage

```
EAMM(
  numsim,
  group,
  repl,
  fixed = c(0, 1, 0),
  VI = seq(0.05, 0.95, 0.05),
  VS = seq(0.05, 0.5, 0.05),
  CoIS = 0,
  relIS = "cor",
  n.X = NA,
  autocorr.X = 0,
  X.dist = "gaussian",
  intercept = 0,
```

```
  heteroscedasticity = c("null"),
  mer.sim = TRUE,
  mer.model = NULL
)
```

Arguments

`numsim`	number of simulation for each step
`group`	number of group
`repl`	number of replicates per group
`fixed`	vector of lenght 3 with mean, variance and estimate of fixed effect to simulate. Default: `c(0, 1, 0)`
`VI`	variance component of intercept. Could be specified as a vector. Default: `seq(0.05, 0.95, 0.05)`
`VS`	Variance component of the slope or IxE. Could be specified as a vector. Default: `seq(0.05, 0.5, 0.05)`
`CoIS`	value of correlation or covariance between random intercept and random slope. Default: 0
`relIS`	"cor" or "cov" set the type of relation give in CoIS. By default the relation is set to correlation
`n.X`	number of different values to simulate for the fixed effect (covariate). If `NA`, all values of X are independent between groups. If the value specified is equivalent to the number of replicates per group, `repl`, then all groups are observed for the same values of the covariate. Default: `NA`
`autocorr.X`	correlation between two successive covariate value for a group. Default: `0`
`X.dist`	specify the distribution of the fixed effect. Only "gaussian" (normal distribution) and "unif" (uniform distribution) are accepted actually. Default: `"gaussian"`
`intercept`	a numeric value giving the expected intercept value. Default: 0
`heteroscedasticity`	a vector specifying heterogeneity in residual variance across X. If `c("null")` residual variance is homogeneous across X. If `c("power",t1,t2)` models heterogeneity with a constant plus power variance function. Letting v denote the variance covariate and $\sigma^2(v)$ denote the variance function evaluated at v, the constant plus power variance function is defined as $\sigma^2(v) = (\theta_1 + \|v\|^{\theta_2})^2$, where θ_1, θ_2 are the variance function coefficients. If `c("exp",t)`,models heterogeneity with an exponential variance function. Letting v denote the variance covariate and $\sigma^2(v)$ denote the variance function evaluated at v, the exponential variance function is defined as $\sigma^2(v) = e^{2*\theta*v}$, where θ is the variance function coefficient. Default:"Null"
`mer.sim`	Use the simluate.merMod function to simulate the data. Potentially faster for large dataset but more restricted in terms of options
`mer.model`	Simulate the data based on a existing data and model structure from a lmer object. Should be specified as a list of 3 components: a mer object fitted via lmer, an environmental covariate for which to test the random slope, a random effect (e.g. `list(fm1,"Days","Subject")`

Details

P-values for random effects are estimated using a log-likelihood ratio test between two models with and without the effect. Power represent the percentage of simulations providing a significant p-value for a given random structure. Residual variance, e, is calculted as 1-VI.

Value

data frame reporting estimated P-values and power with CI for random intercept and random slope

See Also

[PAMM()], [SSF()] for other simulations [plot.EAMM()] for plotting output

Examples

```
## Not run:
ours <- EAMM(
  numsim = 10, group = 10, repl = 4, fixed = c(0, 1, 1),
  VI = seq(0.1, 0.3, 0.05), VS = seq(0.05, 0.2, 0.05)
)
plot(ours, "both")

(fm1 <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy))
ours2 <- EAMM(
  numsim = 10,
  mer.model = list(model = fm1, env = "Days", random = "Subject"),
  VI = seq(0.3, 0.5, 0.1), VS = seq(0.05, 0.2, 0.05)
)
plot(ours2, "both")

## End(Not run)
```

`PAMM` *Simulation function to assess power of mixed models*

Description

Given a specific varaince-covariance structure for random effect, the function simulate different group size and assess p-values and power of random intercept and random slope

Usage

```
PAMM(
  numsim,
  group,
  repl,
  randompart,
```

```
    fixed = c(0, 1, 0),
    n.X = NA,
    autocorr.X = 0,
    X.dist = "gaussian",
    intercept = 0,
    heteroscedasticity = c("null"),
    ftype = "lmer",
    mer.sim = FALSE
)
```

Arguments

`numsim`	number of simulation for each step
`group`	number of group. Could be specified as a vector
`repl`	number of replicates per group . Could be specified as a vector
`randompart`	vector of lenght 4 or 5, with 1: variance component of intercept, VI; 2: variance component of slope, VS; 3: residual variance, VR; 4: relation between random intercept and random slope; 5: "cor" or "cov" determine if the relation 4 between I ans S is a correlation or a covariance. Default: `"cor"`
`fixed`	vector with mean, variance and estimate of fixed effect to simulate. Default: `c(0, 1, 0)`
`n.X`	number of different values to simulate for the fixed effect (covariate). If `NA`, all values of X are independent between groups. If the value specified is equivalent to the number of replicates per group, `repl`, then all groups are observed for the same values of the covariate. Default: `NA`
`autocorr.X`	correlation between two successive covariate value for a group. Default: `0`
`X.dist`	specify the distribution of the fixed effect. Only "gaussian" (normal distribution) and "unif" (uniform distribution) are accepted actually. Default: `"gaussian"`
`intercept`	a numeric value giving the expected intercept value. Default:0
`heteroscedasticity`	a vector specifying heterogeneity in residual variance across X. If `c("null")` residual variance is homogeneous across X. If `c("power",t1,t2)` models heterogeneity with a constant plus power variance function. Letting v denote the variance covariate and $\sigma^2(v)$ denote the variance function evaluated at v, the constant plus power variance function is defined as $\sigma^2(v) = (\theta_1 + \|v\|^{\theta_2})^2$, where θ_1, θ_2 are the variance function coefficients. If `c("exp",t)`,models heterogeneity with an exponential variance function. Letting v denote the variance covariate and $\sigma^2(v)$ denote the variance function evaluated at v, the exponential variance function is defined as $\sigma^2(v) = e^{2*\theta*v}$, where θ is the variance function coefficient.
`ftype`	character value "lmer", "lme" or "MCMCglmm" specifying the function to use to fit the model. Actually "lmer" only is accepted
`mer.sim`	simulate the data using simulate.merMod from lme4. Faster for large sample size but not as flexible.

Details

P-values for random effects are estimated using a log-likelihood ratio test between two models with and without the effect. Power represent the percentage of simulations providing a significant p-value for a given random structure

Value

data frame reporting estimated P-values and power with CI for random intercept and random slope

\@seealso [EAMM()], [SSF()], [plot.PAMM()]

Examples

```
## Not run:
ours <- PAMM(numsim = 10, group = c(seq(10, 50, 10), 100),
             repl = c(3, 4, 6),
             randompart = c(0.4, 0.1, 0.5, 0.1), fixed = c(0, 1, 0.7))
plot(ours,"both")

## End(Not run)
```

`plot.EAMM` *Graphic output of the EAMM function*

Description

Provide graphic interpretation of the simulation results

Usage

```
## S3 method for class 'EAMM'
plot(x, graphtype = "both", vi, vs, fun3D = "wireframe", ...)
```

Arguments

`x`	an EAMM object
`graphtype`	"VI", "VS", or "both" "VI" give graphs with varying variance component of intercept and with a fixed variance component for slope specified in vs argument "VS" give graphs with varying variance component for slope and with a fixed variance component of intercept specified in vi argument "both" 3-D plot see also fun3D argument
`vi`	VI for which plots the output. Necessary for "VS" type of graph
`vs`	VS for which plots the output. Necessary for "VI" type of graph
`fun3D`	plot function used to plot the 3D graph. "wireframe" uses lattice, "persp" uses graphics, and "open3d" uses rgl.
`...`	potentially further arguments to pass to methods

Examples

```
## Not run:
  ours <- EAMM(numsim=10, group=10, repl=4,
               VI=seq(0.1,0.95,0.05), VS=c(0.05,0.1) )
  plot(ours, "both")
  plot(ours, "VI",vs=0.1)
  plot(ours,"VS",vi=0.2)

## End(Not run)
```

`plot.PAMM` *Graphic output of the PAMM function*

Description

Provide graphic interpretation of the simulation results

Usage

```
## S3 method for class 'PAMM'
plot(x, graphtype = "both", nbgroup, nbrepl, fun3D = "wireframe", ...)
```

Arguments

`x`	a PAMM object
`graphtype`	"group", "repl" or "both" "group" give graphs with varying number of ID and with a fixed number of replicates specified in nbrepl "repl" give graphs with varying number of replicates and with a fixed number of ID specified in nbgroup "both" 3-D plot. see also fun3D argument. Note: useful only with multiple group size and multiple number of replicates.
`nbgroup`	number of group for which plots the output. Necessary for "repl" type of graph
`nbrepl`	number of replicates for which plots the output. Necessary for "group" type of graph
`fun3D`	plot function used to plot the 3D graph. "wireframe" uses lattice, "persp" uses graphics and "open3d" uses rgl
`...`	potentially further arguments to pass to methods

Details

Parameters phi, theta, ticktype ("simple" or "detailed"), nticks, nbcol from [persp()] function could also be specified for 3D plots. In addition, color schemes ("grey", "cm.colors" and "rainbow") and coltype ("restricted" or "0-1") parameters could also be specified for 3D plots.

Examples

```
## Not run:
  ours <- PAMM(numsim=10,group=c(seq(10,50,10),100),repl=c(3,4,6),
           randompart=c(0.4,0.1,0.5,0.1),fixed=c(0,1,0.7))
  plot(ours, "both")
  plot(ours, "group",nbrepl=4)
  plot(ours,"repl",nbgroup=20)

## End(Not run)
```

`plot.SSF` *Graphic output of the PAMM function*

Description

Provide graphic interpretation of the simulation results

Usage

```
## S3 method for class 'SSF'
plot(x, ...)
```

Arguments

`x`	an SSF object
`...`	potentially further arguments to pass to methods

Examples

```
## Not run:
   oursSSF <- SSF(50,100,10,c(0.4,0.1,0.6,0))
   plot(oursSSF)

## End(Not run)
```

`SSF` *Simulation function to assess power of mixed models*

Description

Given a specific total number of observations and variance-covariance structure for random effect, the function simulates different association of number of group and replicates, giving the specified sample size, and assess p-values and power of random intercept and random slope

Usage

```
SSF(
  numsim,
  tss,
  nbstep = 10,
  randompart,
  fixed = c(0, 1, 0),
  n.X = NA,
  autocorr.X = 0,
  X.dist = "gaussian",
  intercept = 0,
  exgr = NA,
  exrepl = NA,
  heteroscedasticity = c("null")
)
```

Arguments

`numsim`	number of simulation for each step
`tss`	total sample size, nb group * nb replicates
`nbstep`	number of group*replicates associations to simulate
`randompart`	vector of lenght 4 or 5 with 1: variance component of intercept, VI; 2: variance component of slope, VS; 3: residual variance, VR; 4: relation between random intercept and random slope; 5: "cor" or "cov" determine id the relation between I ans S is correlation or covariance, set to `"cor"` by default
`fixed`	vector of lenght 3 with mean, variance and estimate of fixed effect to simulate
`n.X`	number of different values to simulate for the fixed effect (covariate). If `NA`, all values of X are independent between groups. If the value specified is equivalent to the number of replicates per group, `repl`, then all groups are observed for the same values of the covariate. Default: `NA`
`autocorr.X`	correlation between two successive covariate value for a group. Default: `0`
`X.dist`	specify the distribution of the fixed effect. Only "gaussian" (normal distribution) and "unif" (uniform distribution) are accepted actually. Default: `"gaussian"`
`intercept`	a numeric value giving the expected intercept value. Default:0

`exgr`	a vector specifying minimum and maximum value for number of group. Default:`c(2,tss/2)`
`exrepl`	a vector specifying minimum and maximum value for number of replicates. Default:`c(2,tss/2)`
`heteroscedasticity`	a vector specifying heterogeneity in residual variance across X. If `c("null")` residual variance is homogeneous across X. If `c("power",t1,t2)` models heterogeneity with a constant plus power variance function. Letting v denote the variance covariate and $\sigma^2(v)s2(v)$ denote the variance function evaluated at v, the constant plus power variance function is defined as $\sigma^2(v) = (\theta_1 + \|v\|^{\theta_2})^2 s2(v) = (t1 + \|v\|^t 2)^2$, where $\theta_1, \theta_2 t1, t2$ are the variance function coefficients. If `c("exp",t)`,models heterogeneity with an exponential variance function. Letting v denote the variance covariate and $\sigma^2(v)s2(v)$ denote the variance function evaluated at v, the exponential variance function is defined as $\sigma^2(v) = e^{2*\theta*v} s2(v) = exp(2*t*v)$, where θ is the variance function coefficient.

Details

P-values for random effects are estimated using a log-likelihood ratio test between two models with and without the effect. Power represent the percentage of simulations providing a significant p-value for a given random structure

Value

data frame reporting estimated P-values and power with CI for random intercept and random slope

See Also

PAMM(), EAMM() for other simulation functions plot.SSF() for plotting

Examples

```
## Not run:
oursSSF <- SSF(10, 100, 10, c(0.4, 0.1, 0.6, 0))
plot(oursSSF)

## End(Not run)
```

Index