Package: powerutilities (via r-universe)

October 21, 2026

Title Power Analysis for Linear and Generalized Linear Mixed Models

Description Prospective power analysis and design planning using the ``plug-in'' approach described by Piepho et al. (2022) <doi:10.1017/S0021859622000466> and Stroup, Ptukhina and Garai (2024) <doi:10.1201/9780429092060-24>. Expected treatment means and the proposed experimental design are encoded in an artificial data set, and a model is ``fit'' to it with random effect variances, correlations and residual dispersion fixed at user-specified values rather than being estimated. The resulting standard errors are used to compute the power of F-tests and user-defined contrasts, as well as Type S (sign) and Type M (magnitude) error rates (Gelman and Carlin, 2014 <doi:10.1177/1745691614551642>). Models are fit with 'glmmTMB', so the approach accommodates non-normal responses, heteroscedasticity, nested and crossed random effects, correlated random slopes, and structured covariances such as AR(1) and spatial correlation. Helper functions convert standard deviations and correlations into the internal parameterization of 'glmmTMB' and report the order in which random effect parameters must be supplied. Denominator degrees of freedom can be computed by the Kenward-Roger, residual or asymptotic methods (or can be specified by users), and contrasts are specified through 'emmeans'. Vignettes cover estimated marginal means, contrast specification, and worked examples for common agricultural designs.

Version 0.7.1

Depends R (>= 4.1.0)

Imports dplyr, emmeans, glmmTMB (>= 1.1.14), pbkrtest, reformulas,retrodesign, rlang

Suggests agridat, knitr, rmarkdown, testthat (>= 3.0.0), tidyr

License BSD_3_clause + file LICENSE

URL https://github.com/SimonShamusRiley/powerutilities/tree/main

BugReports https://github.com/SimonShamusRiley/powerutilities/issues

Encoding UTF-8

VignetteBuilder knitr

LazyData true

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

NeedsCompilation no

Author Simon Riley [aut, cre] (ORCID: <https://orcid.org/0000-0002-4833-2026>)

Maintainer Simon Riley <ssriley@protonmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-10-21 19:29:17 UTC

RemoteUrl https://github.com/cran/powerutilities

RemoteRef HEAD

RemoteSha 1c53b314275894109d7f53e64152a5db043c47e4

Contents

barley 2
extract_disp 3
greenfeeds 4
knit_print.powertable 4
power_contrast 5
power_ftest 6
print.powertable 7
print.retermslist 8
set_glmm 8
theta_finder 9

Index **12**

`barley` *Effect of Nitrolime and Superphosphate on Barley Yields*

Description

Data on barley yield for use in interaction contrast example.

Usage

```
barley
```

Format

`barley`:
A data frame with 20 rows and 4 columns:

Rep Factor. The replicate number

Nitrolime Factor. The nitrolime application rate.

Superphosphate Factor. The superphosphate application rate.

Yield Numeric. The yield in kg/ha.

Source

Saville, D. J., & Wood, G. R. (1991). Statistical Methods: The Geometric Approach. Springer Science & Business Media.

`extract_disp` *Extract the possibly fixed residual dispersion parameter*

Description

This is a helper function used for getting `sigma()` when it is not estimated but fixed.

Usage

```
extract_disp(mod, ...)
```

Arguments

`mod`	A glmmTMB model.
`...`	Ignored.

Value

Numeric. The value of the (estimated or specified) residual dispersion in the model.

Examples

```
# Create synthetic data set:
nrep = 8 # Number of replicates
nfac = 2 # Number of treatments
dat = expand.grid(Rep = factor(1:nrep),
                  Trt = factor(LETTERS[1:nfac]))
dat$Y = ifelse(dat$Trt == 'A', 0, 3)

# "Fit" the model while setting residual standard deviation to 2:
mod = set_glmm(Y ~ Trt, data = dat, disp = 2)

# Extract residual SD:
extract_disp(mod)
```

greenfeeds *Fodder Beet Yields*

Description

Data on fodder beet yields for use in class contrasts example.

Usage

```
greenfeeds
```

Format

`greenfeeds`**:**

A data frame with 32 rows and 3 columns:

Treatment Factor. Beet variety.

Rep Factor. The replicate number.

Yield Numeric. Beet yield in tons/ha ...

Source

Saville, D. J., & Wood, G. R. (1991). Statistical Methods: The Geometric Approach. Springer Science & Business Media.

knit_print.powertable *Printing powertables with knitr*

Description

Provides a method for outputting powertables as html

Usage

```
## S3 method for class 'powertable'
knit_print(x, ...)
```

Arguments

`x`	A `powertable` object to be printed
`...`	Unused.

Value

Invisibly returns `x`

power_contrast *Power of Contrasts Performed on Models Fit with glmmTMB*

Description

This function calculates the power of contrasts.

Usage

```
power_contrast(emm, contr_list, ddf = NULL, alpha = 0.05, n_sims = 10000, ...)
```

Arguments

emm	An emmGrid object associated with a glmmTMB model.
contr_list	A (named) list of contrast specifications.
ddf	Either a method for calculating denominator degrees of freedom (currently supported options are "df.residual", "asymptotic" and "kenward-roger"), a numeric value, or NULL (the default), in which case a method is selected based on the model type.
alpha	Numeric. The nominal type I error rate. Defaults to 0.05.
n_sims	Numeric. The number of simulations to use for calculating type M error rate. If set to zero, the closed form, asymptotic calculations are used.
...	Other arguments passed to emmeans::contrast.

Value

A powertable object: a data frame with one row per contrast (or per contrast within each level of any by variables), containing:

Contrast The contrast name, taken from contr_list.

By variables If emm has by variables, one column for each, identifying the group the contrast was computed within.

Estimate The expected value of the contrast, on the link scale.

SE Its standard error under the specified design and variance parameters.

LCL, UCL Lower and upper confidence limits; present only if confidence intervals are requested via infer.

NumDF, DenDF Numerator (always 1) and denominator degrees of freedom for the test.

Fval The expected F statistic (the squared t or z ratio).

Fcrit The critical value of F at significance level alpha.

Pval The expected p-value.

Power The probability of rejecting the null hypothesis of a zero contrast, given the expected estimate.

TypeS The probability that a statistically significant estimate has the wrong sign (Gelman and Carlin, 2014).

TypeM The expected ratio of the absolute value of a statistically significant estimate to the true value (the exaggeration ratio). Inf when the true contrast is zero.

The alpha and ddf attributes record the significance level and the denominator degrees-of-freedom method used.

Examples

```
library(emmeans)

# Create synthetic data set:
nrep = 8 # Number of replicates
nfac = 2 # Number of treatments
dat = expand.grid(Rep = factor(1:nrep),
                  Trt = factor(LETTERS[1:nfac]))
dat$Y = ifelse(dat$Trt == 'A', 0, 3)

# "Fit" the model while setting residual standard deviation to 2:
mod = set_glmm(Y ~ Trt, data = dat, disp = 2)

# Create emmeans object:
emm = emmeans(mod, ~ Trt)

# Define contrasts (here there is only one):
contr = list('A-B' = c(1, -1))

# Calculate power of the contrast
power_contrast(emm, contr)
```

---

power_ftest *Statistical Power of F-tests Performed on Models Fit with glmmTMB*

Description

This function calculates the power of F-tests for each fixed effect term in the model using one of several methods for determining the denominator degrees of freedom.

Usage

```
power_ftest(mod, ddf = NULL, alpha = 0.05, ...)
```

Arguments

mod	A glmmTMB model

ddf	Either a method for calculating denominator degrees of freedom (currently supported options are "df.residual", "asymptotic" and "kenward-roger"), a numeric vector, or NULL (the default), in which case a method is selected based on the model type.
alpha	The nominal type I error rate. Defaults to 0.05.
...	Other values passed to emmeans.

Value

A powertable object: a data frame with one row per fixed-effect term and columns Term, NumDF, DenDF, Fval (expected F statistic), Fcrit (critical value at alpha), Pval (expected p-value) and Power. Attributes alpha and ddf record the significance level and the degrees-of-freedom method used.

Examples

```
# Create synthetic data set:
nrep = 8 # Number of replicates
nfac = 2 # Number of treatments
dat = expand.grid(Rep = factor(1:nrep),
                  Trt = factor(LETTERS[1:nfac]))
dat$Y = ifelse(dat$Trt == 'A', 0, 3)

# "Fit" the model while setting residual standard deviation to 2:
mod = set_glmm(Y ~ Trt, data = dat, disp = 2)

# Calculate power
power_ftest(mod)
```

---

print.powertable *Print the results of a power analysis*

Description

This function controls the formatting when printing powertable objects.

Usage

```
## S3 method for class 'powertable'
print(x, digits = 1, pdigits = getOption("pdigits", default = 4), ...)
```

Arguments

x	A powertable object
digits	Integer. The number of digits to print for all columns other than the p-value and power.
pdigits	Integer. The number of digits to print for the p-value and power columns
...	Unused

Value

The input `x` (a `powertable` object), returned invisibly. Called for its side effect of printing the table with p-values, power and estimates rounded for display, followed by the degrees-of-freedom method and significance level used.

`print.retermslist` *Printing retermslist objects*

Description

Provides a method formatting and displaying retermslist objects

Usage

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

Arguments

`x`	An `retermslist` object, as returned by `theta_finder()`.
`...`	Unused.

Value

The input `x` (a `retermslist` object), returned invisibly. Called for its side effect of printing one table per random-effects term, listing the grouping factor, the name of each random effect, and the position at which each standard deviation and correlation must be supplied in `re_terms` when calling `set_glmm()`.

`set_glmm` *Fit a glmmTMB model with fixed random effects terms*

Description

This is a wrapper to help with fixing random effects terms in glmmTMB model for use in subsequent power analysis. In particular, it assists and converting random effect standard deviation and correlations into the log standard deviation and scaled log cholesky factors that glmmTMB expects.

Usage

```
set_glmm(formula, data, re_terms = NULL, disp = NULL, REML = TRUE, ...)
```

Arguments

formula	A two-sided model formula.
data	A data.frame.
re_terms	A vector of random effects standard deviations and, possibly, correlations.
disp	The value at which to fix the residual dispersion.
REML	Logical. Whether to use REML estimation (default) or, alternatively, ML.
...	Other arguments passed to glmmTMB.

Value

A glmmTMB object.

Examples

```
# Create synthetic data set:
nrep = 8 # Number of replicates
nfac = 2 # Number of treatments
dat = expand.grid(Rep = factor(1:nrep),
                  Trt = factor(LETTERS[1:nfac]))
dat$Y = ifelse(dat$Trt == 'A', 0, 3)

# "Fit" the model while setting residual standard deviation to 2:
mod = set_glmm(Y ~ Trt, data = dat, disp = 2)

# Confirm it worked:
sigma(mod)
```

---

theta_finder — *Identify how random effects terms will be ordered in the fitted glmmTMB model*

Description

When performing a "plug-in" power analysis, the values for the random effects must be supplied in the order in which they are saved internally in the glmmTMB model. It is not always clear (especially where there are correlated random slopes and intercepts) just what this ordering is. This helper function takes a model formula and a data set and outputs a table enumerating all the implied random effects and their ordering.

Usage

```
theta_finder(formula, data, ...)
```

Arguments

`formula`	A model formula (containing random effects).
`data`	A data frame containing all the columns included in `formula`.
`...`	Other arguments passed to `glmmTMB`.

Value

A 'retermslist' object numbering each of the random effect variances and correlations implied by the model formula.

Examples

```
library(dplyr)

# Following the original analysis by Cornelius and Archbold, row x spacing
# is treated as main-plots, and stock within main-plots as sub-plots.
apple_des = agridat::archbold.apple |>
  mutate(across(rep:gen, factor),
         mp = interaction(row, spacing, sep = '-'),
         sp = interaction(row, spacing, stock, sep = '-'))

# If we opt to treat blocks as random, then here is the
# appropriate model for a split-split-plot RCBD:
ssp_mod = ~ spacing*stock*gen + (1|rep/mp/sp)

# The first theta value corresponds to to the standard deviation among
#  sub-plots, the second theta value corresponds to the of the standard deviation
#  main-plots and the third theta value corresponds to the standard deviation
#  among blocks:
theta_finder(formula = ssp_mod, data = apple_des)

# Purely for the sake of illustration, lets add hypothetical random slopes
# for soil pH at subplot level. In the data, the values don't need to be
# well simulated or even meaningful:
apple_des2 = apple_des |>
  mutate(pH = 7)

# Note that mp and sp are already explicitly defined in terms of their
# nesting structure, so defining them here is terms of their interactions
# isn't really necessary
ssp_cov_mod = ~ spacing*stock*gen + (1|rep) + (1|rep:mp) + (pH + 1|rep:mp:sp)

# Not one but two new theta values have been added to the model: we also
# have the correlation between the pH slope and split-plot intercepts.
theta_finder(formula = ssp_cov_mod, data = apple_des2)

# Finally, be aware that redundant random effects terms, which would simply
# be consolidated when fitting a real model, are problematic here. In this example,
# rep is included twice:
bad_mod = ~ spacing*stock*gen + (1|rep) + (1|rep/mp)
```

```
try(theta_finder(formula = bad_mod, data = apple_des))
```

Index

∗ **datasets**
    barley, 2
    greenfeeds, 4

barley, 2

extract_disp, 3

greenfeeds, 4

knit_print.powertable, 4

power_contrast, 5
power_ftest, 6
print.powertable, 7
print.retermslist, 8

set_glmm, 8
set_glmm(), *8*

theta_finder, 9
theta_finder(), *8*