# Introduction to mixedpower

February 23, 2020

**Title:** Simulation-based power analyses for (generalized) linear mixed models

**Version**: 2.0

**Authors:** Kumle, Võ & Draschkow (2018)

**Maintainer:** Leah Kumle leahkumle@gmail.com



**URL:** https://github.com/DejanDraschkow/mixedpower

**Description:** Perform a priori simulation-based power analyses in (generalized) linear mixed-effect models for a range of different design parameters (e.g. levels of different random factors or expected effect sizes). All simulations are based on models fitted with lme4 (Bates, Mächler, Zurich, Bolker, & Walker, 2015).

**Imports:** lme4, ggplot2, doParallel, foreach, reshape

**Supporting Information:** Exemplary analyses using mixedpower can be found at https://lkumle.github.io/power_notebooks/ and more detail on the theoretical background can be found in Kumle, Võ & Draschkow (in preparation).

## Table of Contents

## Simulation-based power analyses

Being able to estimate power is incredibly important in order to plan adequately powered studies as power is closely linked to reliability and replicability of empirical findings. Classical solutions to power analysis work with analytical formulas. However, (generalized) linear mixed models ((G)LMMs) are often too complex to be solved analytically and therefore require a different approach. A flexible and more intuitive alternative to analytic power solutions are simulation-based power analyses (Brysbaert & Stevens, 2018; Thomas & Juanes, 1996). In simple terms, one basic question behind power analyses is: "Suppose there really is an effect of a certain size and I run my experiment one hundred times - how many times will I get a statistically significant result?" (Coppock, 2013). As it is possible to simulate the outcome of an experiment, power can be calculated based on the proportion of significant simulations to all simulations (Johnson et al., 2015; Thomas & Juanes, 1996). The shared principle of all simulation-based power analyses solutions included in mixedpower can therefore be broken down into the following steps:

1) simulation of new data sets,
2) analysis of each data set and test for statistical significance, and
3) calculation of the proportion of significant to all simulations.

However, accuracy of the power estimate undoubtedly and heavily depends on the accuracy of our simulation - informing the parameters of the simulation therefore is a critical step. A more detailed discussion of possible theoretical concerns can be found in Kumle, Võ & Draschkow (in preparation).

**Mixedpowe r**

| **mixedpower** | *Simulate power for one varying random effect* |
|---|---|

Description: mixedpower() simulates power for all specified fixed effects and their interactions in a provided model fitted with lme4 (CITE). It allows to vary the levels of one random effect and beta coefficients to estimate power for different designs and effect sizes.

Usage:

```
mixedpower(model, data, fixed_effects, simvar,
           steps, critical_value, n_sim = 1000,
           SESOI = F, databased = T)
```

Arguments:

| | |
|---|---|
| model | a fitted lme4-model for which fixed effects (+ interactions) power should be simulated |
| data | data used to fit the model of interest |
| fixed_effects | variable names of all fixed effects specified in *model* |
| simvar | variable name of random effect that is supposed to vary in simulation |
| steps | steps of *simvar* (e.g. different sample sizes power should be simulated for) |
| critical_value | t (for LMMs) or z (for GLMMs) value used to test significance in the simulated data |
| n_sim | number of repetitions in simulation process (i.e. how many simulations should the estimate be based on?) |
| SESOI | specification of *Smallest effect of interest.* Allows to change the beta coefficients to include a simulation with modified effect sizes by providing a vector with new beta coefficients. If *F*, no modification is made. |
| databased | logical: Include simulation with beta coefficients found in *model* ? |

Value:

Data frame with power estimates for all fixed effects (and their interactions) and all specified steps. Depending on if SESOI and/or databased is included, separate power estimates for both methods are included.

Details:

According to the three general steps in simulation-based power analysis, mixedpower() first simulates *n_sim* new data sets using the simulateDataset() inside the power_simulation() function (see details below). In a second step, the provided *model* is refitted with the new simulated data set. Using the provided *critical_value*, the included fixed effects and their interactions are tested for significance. After all simulations are finished, the proportion of significant to all simulation is computed and stored. This process is repeated for all different specification (i.e. different steps and beta coefficients).

| **R2power** | *Simulate power for two varying random effects* |
|---|---|

Description: R2power simulates power for all specified fixed effects and their interactions in a provided model fitted with lme4 (CITE). It allows to vary the levels of one random effect and beta coefficients while simultaneously changing the level of a second random effect.

Usage:

```
R2power(model, data, fixed_effects, simvar,
        steps, R2var, R2level,critical_value,
        n_sim = 1000, SESOI = F, databased = T)
```

Arguments:

| | |
|---|---|
| model | a fitted lme4-model for which fixed effects (+ interactions) power should be simulated |
| data | data used to fit the model of interest |
| fixed_effects | variable names of all fixed effects specified in *model* |
| simvar | variable name of the first random effect that is supposed to vary over multiple different steps in simulation |

| | |
|---|---|
| steps | steps of *simvar* (e.g. different sample sizes power should be simulated for) |
| R2var | variable name of a second random effect which's level is supposed to be changed in the simulation |
| R2level | level the second random effect (i.e. R2var) should be changed to |
| critical_value | t (for LMMs) or z (for GLMMs) value used to test significance in the simulated data |
| n_sim | number of repetitions in simulation process (i.e. how many simulations should the estimate be based on?) |
| SESOI | specification of *Smallest effect of interest.* Allows to change the beta coefficients to include a simulation with modified effect sizes by providing a vector with new beta coefficients. If *F*, no modification is made. |
| databased | logical: Include simulation with beta coefficients found in *model* ? |

Value:

Data frame with power estimates for all fixed effects (and their interactions) and all specified steps. Depending on if SESOI and/or databased is included, separate power estimates for both methods are included.

Details:

According to the three general steps in simulation-based power analysis and similar to mixedpower(), R2power() first simulates *n_sim* new data sets using the simulateDataset() inside the power_simulation() function (see details below). In a two-step simulation process, the levels of the random effect in *simvar* is simulated first before adapting the data set to the level of the second random effect specified in *R2level.* In a second step, the provided *model* is refitted with the new simulated data set. Using the provided *critical_value*, the included fixed effects and their interactions are tested for significance. After all simulations are finished, the proportion of significant to all simulation is computed and stored. This process is repeated for all different specification (i.e. different steps and beta coefficients).

**multiplotPower** *Plot results*

Description: multiplotPower() plots the results achieved with mixedpower() or R2power(). A separate subplot for each fixed effect and interaction is drawn. Databased and SESOI is combined in one plot for each effect.

Usage:
```
multiplotPower(output_data, ppi = 300, filename = F)
```

Arguments:

| | |
|---|---|
| output_data | data frame returned by mixedpower() or R2power() |
| ppi | resolution of plot |
| filename | allows for customized filename as plot is automatically saved to current working directory. Should be character ending with ".png" |

Value:

PNG-file containing all subplots saved to current working directory.

**simulateDataset** *Simulate new dataset*

Description: simulateDataset() allows to simulate new response values, which are integrated into the original data set while simultaneously changing the level of one random factor if specified accordingly.

Usage:
```
simulateDataset(n_want, data, model,
                    simvar, use_u = F)
```

Arguments:

| | |
|---|---|
| n_want | desired level of specified random effect |
| data | data used to fit the model of interest |
| model | a fitted lme4-model for which fixed effects (+ interactions) power should be simulated |

simvar — variable name of the first random effect that is supposed to be varied

use_u — referring to *use.u* argument in the lme4::simulate.Mermod() function. Logical: if F, random effect values are drawn out of normal distribution. If T, the exact values found in *model* are used for the simulation process. *Use_u* is set to F in the two-step simulation process of R2power() to ensure one data set is simulated with one set of random effect structures.

Value:

Data frame with modified response values and levels in one of the random effects.

Details:

simulateDataset() makes use of lme4::simulate.Mermod() to simulate new response values. Resulting data frame is either extended with additional simulated response values and variable identifiers or subsetted to the required level of the specified random effect.

---

**power_simulation** *loops through multiple simulations*

---

Description: power_simulation() combines multiple repetitions of simulateDataset() into one simulation process while changing parameters to estimate power for different design and random effect specifications.

Usage:

```
power_simulation(model, data, simvar, fixed_effects,
                 critical_value, steps, n_sim,
                 confidence_level,safeguard = F,
                 rnorm = F, R2 = F, R2var, R2level)
```

Arguments:

model — a fitted lme4-model for which fixed effects (+ interactions) power should be simulated

data — data used to fit the model of interest

fixed_effects — variable names of all fixed effects specified in *model*

| | |
|---|---|
| critical_value | t (for LMMs) or z (for GLMMs) value used to test significance in the simulated data |
| simvar | variable name of the first random effect that is supposed to vary over multiple different steps in simulation |
| steps | steps of *simvar* (e.g. different sample sizes power should be simulated for) |
| n_sim | number of repetitions in simulation process (i.e. how many simulations should the estimate be based on?) |
| confidence_level | remaining of a previous mixedpower version. Used to estimate power based on the lower bound of a confidence interval around the beta coefficients. Parameter indicates the widths of this confidence interval. |
| safeguard | logical: Estimate power based on lower bound of confidence interval? (Only used in previous version of mixedpower) |
| Rnorm | Remaining of a previous version of mixedpower. Beta coefficients are randomly drawn from distribution around original beta coefficients. |
| R2 | logical: is current simulation part of a R2power() simulation? |
| R2var | variable name of a second random effect which's level is supposed to be changed in the simulation |
| R2level | level the second random effect (i.e. R2var) should be changed to |

Value:

Power values for all fixed effects and their interactions for the specified simulation parameters.

Details:

Power_simulation() loops though a specified number of simulations (*n_sim*) in parallel and computes power by calculating the relation between significant to all simulations. First, a new data set is simulated using simulateDataset(). Before refitting the model used for simulation, all contrasts specified in the original data are reapplied to the simulated data set. Then, significance for all specified fixed effects and their interactions is assessed and stored. After all repetitions are finished, power is calculated.

## References


Brysbaert, M., & Stevens, M. (2018). Power Analysis and Effect Size in Mixed Effects Models: A Tutorial. *Journal of Cognition*, *1*(1), 1–20. https://doi.org/10.5334/joc.10

Coppock, A. (2013). 10 Things to Know About Statistical Power. Retrieved September 20, 2018, from http://egap.org/methods-guides/10-things-you-need-know-about statistical-power

Johnson, P. C. D., Barry, S. J. E., Ferguson, H. M., & Müller, P. (2015). Power analysis for generalized linear mixed models in ecology and evolution. *Methods in Ecology and Evolution*, *6*(2), 133–142. https://doi.org/10.1111/2041-210X.12306

Thomas, L., & Juanes, F. (1996). The importance of statistical power analysis: An example from Animal Behaviour. *Animal Behaviour*, *52*(4), 856–859. https://doi.org/10.1006/anbe.1996.0232