# simr : an R Package for Power Analysis of Linear Mixed Models by Simulation

## Introduction

### Why we want to do power analysis.

If we are going to invest in the collection of monitoring data, it is important that we know that our design is suitable for making the inferences we need. Does the design have sufficient power to detect what we are trying to find? Are our parameter estimates going to be precise enough for our purposes?

When the design is complicated, e.g. in a mixed-effects model, simulation provides a general way of answering these questions. The simr package provides tools that make it simple to set up and run simulation experiments.

Need to make sure our monitoring programme can find what we’re looking for.

Want our study to be cost effective.

Talk about Field’s stuff on power analysis.

### What simr does.

With simr you can perform a power analysis based on a linear mixed model fitted with the lmer command in the lme4 package. You can also use linear models fitted with lm from base R. This is not strictly necessary, but allow for a shallower learning curve, especially since these models can be fitted much faster than mixed models.

This paper runs through [four?] tutorials. The first uses lm and guides the user through a very simple posthoc power analysis. The second uses lmer and uses some of the more flexible features of the package.

### Why we’re using mixed models.

Linear mixed models allow us to model the random variation among units as well as the effect of explanatory variables [Bolker 2008].

### Why we want to use simulation.

Simulation is a general procedure for determining the properties of a statistical method or design [ref]. With simulation we do not need to find an analytical approach specialised to our particular analysis, and we do not need to rely on an approximation that may not be robust to departures from its assumptions. This can be especially important with mixed models where the sampling distributions of parameter estimates are difficult to work with [ref Bates?].

### Why we need an R package.

Simulation studies can be difficult or time consuming to set up. They would normally involve some degree of programming by the investigator. Depending on their comfort level, this might be beyond their current ability or might simply take longer than they might like.

Usually done with a one-off script [ref simFrame] – coordination btw researchers (replication?)

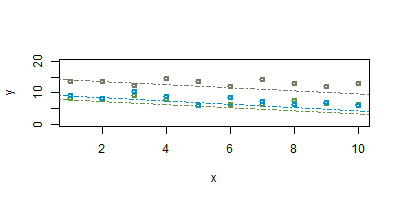
Having a ready-made package available could make power analyses accessible to a wider range of scientists. Investigators who might otherwise be limited to fitting a model in R [ref] could supplement their study with a power analysis without much additional effort. For scientists comfortable with R coding, a turnkey package could save them time and let them focus their efforts on a more advanced analysis.

### Other packages

longpower [ref] uses an analytical approximation [describe, ref] to calculate power for longitudinal studies. [include calculations as an option in simr?]. pamm is a package for doing similar simulations, but with a restricted set of models [ref]. Packages simSummary [ref] and simFrame [ref]

## Tutorial One: Post-hoc power analysis.

The example dataset has response variable y measured at ten levels of the explanatory variable x for three groups g.



We can fit a simple random intercepts model as follows

library(simr) # note that this will also load lme4

fit <- lmer(y ~ x + g, data=example)

Figure [one] shows a scatterplot of the data with these fitted lines. If we were to repeat this experiment, what would be our power to detect a trend this size?

power(fit)

[explanation of the result; where it comes from, what it means]

This level of power might be acceptable in many applications. But suppose that collecting data on many levels of x is costly – we might want to collect only as much data as we need to reach a certain level of statistical power. For this we would need to calculate a power curve, which tells us the trade-off between sample size and power.

## Tutorial Two: Calculate a power curve

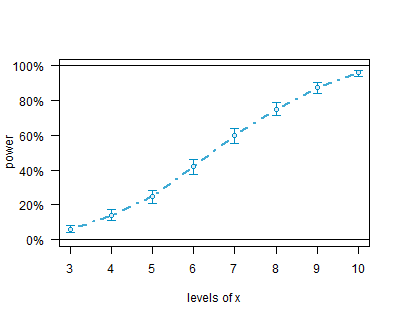
Calculating and plotting a power curve in simr is simple:

pc <- powercurve(fit)

plot(pc)

Note that this may take some time; we are analysing [R] simulated datasets, and for each of those simulated datasets we are fitting the model to 8 different subsets (we are varying the sample size from 3—10 levels of x).

The result is shown in figure [two].



## Tutorial Three: Specify the sample and effect sizes.

The first two analyses made heavy use of the default settings in simr to keep things as simple as possible. For example, the simulated trend defaults to the trend estimated in the fitted model. Often we will have a specific value for an ecologically significant effect, and we are interested in the power to detect and effect of that size. We can access the fixed effects in an linear mixed model with:

fixef(fit)

If we want a specific fixed effect, say the effect for x we use:

fixef(fit)[‘x’]

Suppose that our ecologically significant effect size for x is -0.1. simr uses the (hopefully) obvious idiom to change the size of fixed effects:

fixef(fit)[‘x’] <- -0.1

We can now calculate a power curve for our modified fitted model the same way we did for [tutorial two]:

pc3a <- powercurve(fit)

plot(pc3a)

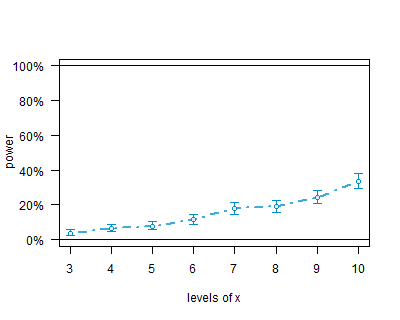
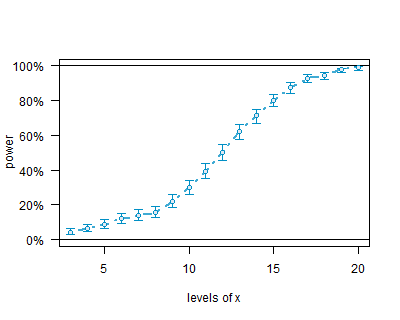
Again, this will take some time. However you can see the results now in figure [3a]. This analysis shows us that we have insufficient power at this effect size for any of the sample sizes considered. To get a better picture of the trade-off between power and sample size we need to increase the number of levels of x, which we can do using the extend command:

fit <- extend(fit, along=’x’, n=20)

pc3b <- powercurve(fit)

plot(pc3b)

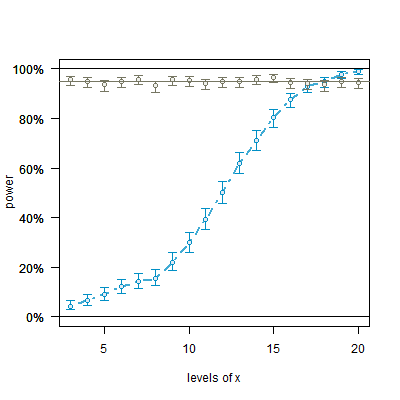
Figure [3b] shows the power curve for a larger range of levels of x (3—20).



## Tutorial Four: Include Type I error calculations.

pc4 <- powercurve(fit, null=TRUE)

plot(pc4)



## Further Work.

As currently implemented, simr uses data from a pilot study to inform the power analysis, with the structure of the pilot data providing defaults for the simulation settings. Future versions will also include the ability to create data sets from scratch; this will not require a pilot study, but will require some domain expertise to select sensible parameters.

Version 1.0 of simr is designed for linear mixed models using lmer in lme4. The next version will add support for generalised linear models (glm in base) and generalised linear mixed models (glmm in lme4). At some point tools will be added to make it simple to create interfaces to arbitrary R packages.

## Notes: Which Dataset to Use?

Simulated dataset: conceptually simple, allows the package to be the focus of the paper.

Real dataset: allows practical scientists to get a handle on what’s actually happening. But we have to deal with introducing the data and with any peculiarities; this all takes away from the main focus of the paper.

Simulated “Real” Dataset: Combines advantages of both approaches. We generate a clean simulated dataset; this means we spend no time explaining data cleaning assumptions. But the variables are given meaningful names so that people can develop an intuitive grasp of what’s going on.