

# ACRM 2023 Longitudinal Data Analysis Workshop

Keith Lohse, PhD, PStat, and Allan Kozlowski, PhD, BsC (PT)

## Practical Session 4: Simulation-based Approaches to Statistical Power in Mixed-Effect Models

### 1.A Refresher about Statistical Power

In previous sessions, we explored how to visualize longitudinal data, how to build mixed-effect regression models, and how to conduct statistical significance tests in the context of linear and non-linear mixed-effect regression. In this session, we will focus on statistical power as a complement to null-hypothesis testing.

As before, we will use several packages related to data management, visualization, and mixed-effect models.

```
# Loading the essential libraries.
library("tidyverse")
library("lme4")
library("MASS")
library("car")
library("lmerTest")
library("nlme")
```

If you have not already installed these packages, you will need to use the `install.packages()` function first. This can take some time and will require an internet connection.

```
# If these packages are not installed already, run the following code:
install.packages("tidyverse")
install.packages("lme4")
install.packages("car")
install.packages("lmerTest")
install.packages("nlme")
```

Before we get into the details of hypothesis testing with longitudinal data, let's briefly refresh the concept of statistical power. Calculating statistical power requires us to first figure out what would happen under the *null hypothesis*. In this case we assume that the true mean difference is zero ( $H_0: \delta=0$ ). If the null hypothesis were true, we would expect a distribution of mean differences like the one in blue. The distribution is centered on 0, but the difference we observe will vary from study to study. Specifically, 95% of the null distribution is between the vertical black lines, putting 2.5% in each tail. If a mean difference falls outside of those black lines (i.e.,  $p < 0.05$  if the null is true), then we would call the result “statistically significant” and reject the null hypothesis. We use  $\alpha=0.05$  as a cutoff by convention, but this limit could be set to any value and conventions will vary by scientific discipline. Critically though, by setting this  $\alpha$  threshold allows us to control the Type I error rate. That is, if we set  $\alpha=0.05$ , that means we will only make a mistake and declare an effect statistically significant 5% time when the null is true (i.e., we will make “false alarms” 5% of the time).

However, we also need to consider what might happen if the null is *not* true. For instance, what if the true mean difference was  $\delta=4$  instead of  $\delta=0$ ? If this *alternative hypothesis* was true, then we would expect a distribution of mean differences like the one in red. Under this assumption, 50% of our results are above the statistical significance threshold and 50% are below the threshold. Thus, if  $\delta=4$  is true, we will fail to find a statistically significant effect 50% of the time (i.e., we will miss 50% of the time). This miss rate is referred to as the Type II error rate and represented by  $\beta$ .

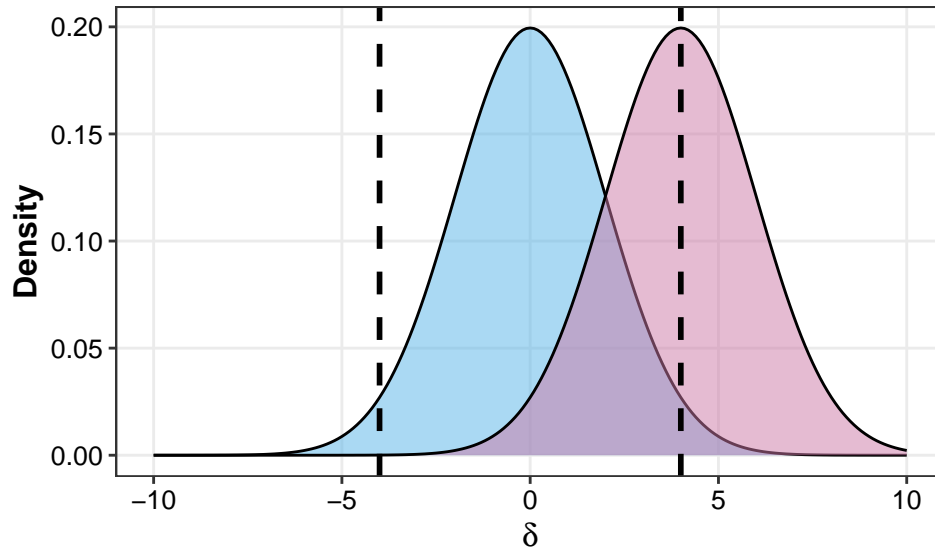
Statistical power is  $1-\beta$ . Put another way, statistical power is the probability of obtaining a statistically significant result if a given alternative hypothesis is true. In the figure, statistical power is not very good, because if the true mean difference is  $\delta=4$ , we will only correctly reject the null hypothesis 50% of the time!

```
differences <- c(seq(-10, 10, 0.1))
null_dist <- dnorm(x = differences, mean = 0, sd = 2)
alt_dist <- dnorm(x = differences, mean = 4, sd = 2)

DATA <- data.frame(differences, null_dist, alt_dist)

cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#CC79A7", "#F0E442",
               "#555555", "#D55E00", "#0072B2", "#03c478", "#661100", "#f3f319",
               "#222222", "#FBD7A2", "#6699CC", "#99edcc", "#b804a2", "#F9E999")

ggplot(data=DATA, aes(x=differences)) +
  geom_density(aes(y=null_dist), fill=cbPalette[3], stat="identity", alpha=0.5)+
  geom_density(aes(y=alt_dist), fill=cbPalette[5], stat="identity", alpha=0.5)+
  geom_vline(xintercept = c(-4, 4), lty=2, lwd=1, col="black")+
  scale_x_continuous(name = expression(delta)) +
  scale_y_continuous(name = "Density") +
  theme_bw()+
  theme(axis.text=element_text(size=10, color="black"),
        legend.text=element_text(size=10, color="black"),
        legend.title=element_text(size=10, face="bold"),
        axis.title=element_text(size=12, face="bold"),
        plot.title=element_text(size=12, face="bold", hjust=0.5),
        panel.grid.minor = element_blank(),
        strip.text = element_text(size=10, face="bold"),
        legend.position = "none")
```



This figure highlights why we can't solely focus on getting statistically significant results. The alternative hypothesis might very well be true, but if we do not have enough people/observations, then we may lack the statistical power to detect it. Thus, anytime you are designing a study you need to consider statistical power, how high you want statistical power to be, and what changes you might be able to make to improve statistical power. E.g.,

1. Increasing the size of the effect (e.g., giving a drug at a high dose compared to a moderate dose).
2. Reducing the variability in the data (e.g., through inclusion/exclusion criteria or controlling for relevant covariates).
3. Increasing the number of observations (e.g., recruiting more participants and/or taking more observations per participant)

In the context of longitudinal studies, we are commonly most interested in *cross-level interactions*. For instance, did people receiving the experimental treatment change at the same rate as people receiving the control treatment? Or, do people with more severe baseline impairment show similar trajectories compared to people with less severe baseline impairment? These Group x Time interactions can take many forms and we cannot possibly cover them all here. However, we will try to provide some fundamental tools and show the logic of using simulation based approaches to statistical power. Users can then adapt this code to their specific case, changing things like the number of observations, the amount and pattern of missing data, the (non)linearity of their trajectories, and the magnitude of the effects they are predicting.

## 2. A Simulation Approach to Longitudinal Data

BLARG!

```
# set number of individuals
N <- 10

# average intercept and slope
beta0 <- 50
beta1 <- 1
```

```

# true error SD, intercept SD, slope SD, and intercept-slope cor
sigma <- 5
tau0 <- 10
tau1 <- 2
tau01 <- 0.5

# number of possible observations
max_obs <- 5
min_obs <- 4

# simulate MISSING AT RANDOM observations for each individual
set.seed(42)
p <- round(runif(n = N, min = min_obs, max = max_obs))

# simulate observations per person (everyone has 1st observation)
time <- unlist(sapply(p, function(x) c(1, sort(sample(x = 2:max_obs, size = x - 1,
  replace = FALSE)))))

# set up data frame
DATA <- data.frame(id = rep(1:N, times = p), time = time)

# simulate (correlated) random effects for intercepts and slopes
mu <- c(0, 0)
S <- matrix(c(1, tau01, tau01, 1), nrow = 2)
taus <- c(tau0, tau1)
S <- diag(taus) %*% S %*% diag(taus)
U <- mvrnorm(N, mu = mu, Sigma = S)

# simulate (uncorrelated) residuals you can simulate correlated residuals, but
# that takes a bit more work
set.seed(42)
DATA$eij <- rnorm(n = nrow(DATA), mean = 0, sd = sigma)

DATA$yij <- (beta0 + rep(U[, 1], times = p)) + (beta1 + rep(U[, 2], times = p)) *
  DATA$time + DATA$eij

cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#CC79A7", "#F0E442",
  "#555555", "#D55E00", "#0072B2", "#03c478", "#661100", "#f3f319",
  "#222222", "#FBD7A2", "#6699CC", "#99edcc", "#b804a2", "#F9E999")

# Lattice plot of the example data ----
ggplot(data=DATA, aes(x=time, y=yij)) +
  geom_point(shape=16, col="black")+
  geom_line(col="black")+
  stat_smooth(aes(group=id), col="blue", se=FALSE,
    method="lm")+
  scale_x_continuous(name = "Time") +
  scale_y_continuous(name = "Outcome") +
  facet_wrap(~id, ncol=5) +
  theme_bw()+
  theme(axis.text=element_text(size=10, color="black"),
    legend.text=element_text(size=10, color="black"),
    legend.title=element_text(size=10, face="bold"),

```

```
axis.title=element_text(size=10, face="bold"),
plot.title=element_text(size=12, face="bold", hjust=0.5),
panel.grid.minor = element_blank(),
strip.text = element_text(size=10, face="bold"),
legend.position = "none")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

