

LightSpan Power Analysis

Johannes Zauner

Prelude

This document provides a sensitivity analysis of required sample size depending on the outcome of an intervention for Time above threshold 250 lx mel EDI, Time above threshold 1000 lx mel EDI, and Time below threshold 10 lx mel EDI at evening, where evening is defined as the time from civil dusk till midnight.

The data and metric values are taken from Biller et al. 2025. Using the analysis script and data from their analysis, data were extracted with the following command:

```
metrics |>
  filter(metric %in% c("TAT250", "TAT1000", "TBTe10")) |>
  unnest() |>
  filter(site == "malaysia") |>
  select(-c(photoperiod_duration, Photoperiod, Day, site)) |>
  mutate(intervention = 0) |>
  write.csv("input.csv")
```

These data will be considered the pre-condition prior to intervention, indicated through their status of intervention = 0.

Setup

```
library(tidyverse)
```

```
— Attaching core tidyverse packages ————— tidyverse 2.0.0
—
✓ dplyr      1.1.4    ✓ readr      2.1.5
✓ forcats    1.0.0    ✓ stringr    1.5.1
✓ ggplot2    3.5.1    ✓ tibble     3.2.1
✓ lubridate  1.9.4    ✓ tidyr      1.3.1
✓ purrr      1.0.4
— Conflicts —————
tidyverse_conflicts() —
* dplyr::filter() masks stats::filter()
* dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(broom.mixed)
library(simr)
```

```
Loading required package: lme4
Loading required package: Matrix

Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

  expand, pack, unpack

Attaching package: 'simr'

The following object is masked from 'package:lme4':

  getData

The following object is masked from 'package:stringr':

  fixed
```

```
library(cowplot)
```

```
Attaching package: 'cowplot'

The following object is masked from 'package:lubridate':

  stamp
```

```
library(gghighlight)
library(patchwork)
```

```
Attaching package: 'patchwork'

The following object is masked from 'package:cowplot':

  align_plots
```

```
data <- read.csv("input.csv")

set.seed(20250325)

#source all files in function
list.files("functions", full.names = TRUE) |> walk(source)
```

General settings

The critical sample size will be calculated based on a threshold power of:

```
power_level <- 0.8  
sign_level <- 0.05
```

Changes in metrics will be calculated in % increases from baseline

```
change_range <-  
  c(0.1, 0.2, 0.3, 0.4, 0.5)
```

It is further assumed, that not every participant reacts the same to the intervention, thus there is a standard deviation to that increment

```
sd_increment <- 0.20
```

For each increment, a bootstrapped number of datasets are created

```
n_replicates <- 200
```

And the following sample sizes are tested:

```
sample_size <- c(5,10,15, 20, 25, 30, 35, 40, 45, 50)
```

Further, a maximum of days are considered:

```
max_days <- 7
```

We analyse the effect according to this formula:

```
formula <- value ~ intervention + (intervention|Id)
```

Generating the effect-data

In this section we generate the effect data based on our participant level.

```
data_extended <-  
  change_range |>  
  map(\(x) {  
    data |>  
      group_by(metric, Id) |>  
      mutate(  
        # value.intervention = value * (  
        #   1 + x * rnorm(1, mean = 1, sd = sd_increment)  
        # ) |> round(0),  
        # alternative implementation  
        value.intervention = rpois(length(value), mean(value * (  
          1 + x * rnorm(1, mean = 1, sd = sd_increment)
```

```

    )),
    value = rpois(length(value), mean(value)),
  ) |>
  select(-intervention) |>
  pivot_longer(cols = c(value, value.intervention),
               names_to = "intervention") |>
  mutate(intervention = case_when(intervention == "value" ~ 0,
                                intervention == "value.intervention" ~
1))
}) |> list_rbind(names_to = "effect") |>
  mutate(effect = set_names(change_range, nm = 1:length(change_range))
[effect])

```

Generate Bootstraps

In this section we create the bootstrapped datasets

```

Power_data <-
  data$metric |>
  unique() |>
  map(\(x) bootstrap(data_extended, x) |>
    mutate(metric = x)) |>
  list_rbind()

```

boundary (singular) fit: see help('isSingular')

```

Warning: There were 2 warnings in `mutate()`.
The first warning was:
i In argument: `model = list(glmmer(formula, data = data, family = "poisson"))`.
i In row 257.
Caused by warning in `checkConv()`:
! Model failed to converge with max|grad| = 0.00208299 (tol = 0.002, component
1)
i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.

```

Visualizing Power

```

Power_summary <-
Power_data %>%
  unnest(data) %>%
  mutate(Power_reached = power >= power_level) %>%
  group_by(metric, effect) |>
  filter(Power_reached, .preserve = TRUE) %>%
  slice_min(sample_size) %>%
  select(-Power_reached) %>%
  ungroup()

#check which metrics did not reach the threshold
missing <-

```

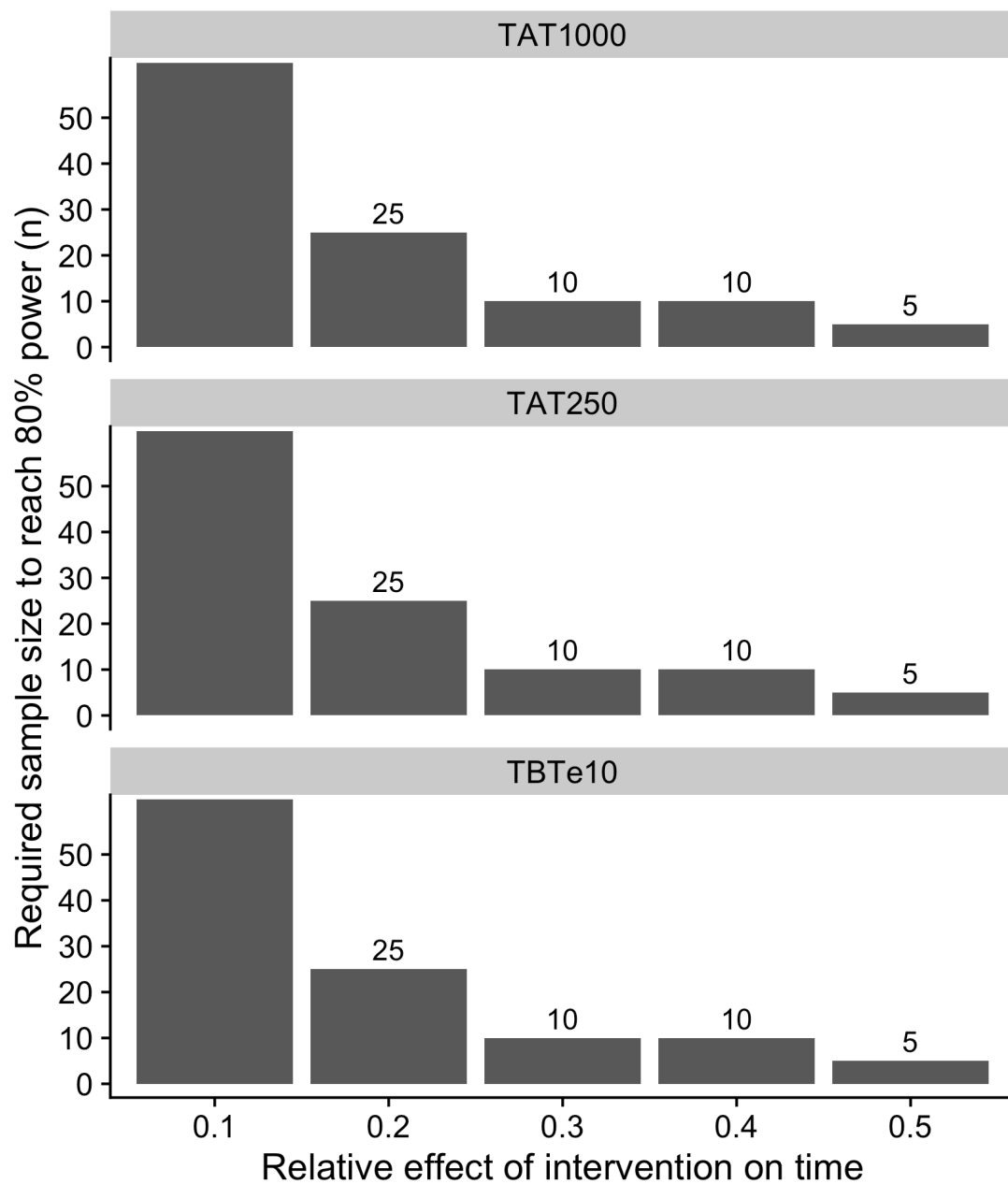
```

anti_join(Power_data, Power_summary, by = join_by(effect, metric)) |>
select(effect, metric)

Power_summary <-
Power_data |>
select(effect, metric) |>
left_join(Power_summary, by = join_by(effect, metric))

Power_summary |>
  mutate(sample_size = case_when(is.na(sample_size) ~ 62, .default =
sample_size)) |>
  ggplot(aes(x = factor(effect), y = sample_size)) +
  geom_col() +
  # scale_x_discrete(breaks = as.character(change_range), label =
scales::label_percent()) +
  facet_wrap(~metric, ncol = 1) +
  coord_cartesian(
    # xlim = c(0.022, 1.1),
    ylim = c(0, max(sample_size)+10)
  ) +
  scale_y_continuous(breaks = seq(from = 0, to = max(sample_size), by = 10))+
  theme_cowplot() +
  labs(x = "Relative effect of intervention on time",
    y = "Required sample size to reach 80% power (n)",
    caption = paste0("Based on sample size iterations of 5 (min. 5), and ",
max_days, " days per participant pre- and post intervention.\nBars without a
number indicate the required sample size is above n=", max(sample_size), ".
\nPower calculation based on ", n_replicates, " per sample size and metric.")
  ) +
  geom_text(aes(label = sample_size, y = sample_size + 2), vjust = 0)

```



size iterations of 5 (min. 5), and 7 days per participant pre- and post intervention.
 Bars without a number indicate the required sample size is above n=50.
 Power calculation based on 200 per sample size and metric.

```
# gghighlight(sample_size <= 50)

ggsave("figures/LightSpan_power.pdf", width = 7, height = 7)

# #apply all metrics to the plot function# #apply all metrics to the plot
function# #apply all metrics to the plot function
# Power_data <-
#   Power_data %>%
```

```

# mutate(plot =
#       map2(data,
#             interaction(metric, effect),
#             plot_power,
#             power_level = power_level, breaks = sample_size))
#
# #create all plots in a grid and save it
# Figure.4 <-
#   Power_data$plot %>%
#   reduce\(x, y) x + y) +
#   plot_annotation(tag_levels = 'A') &
#   theme(plot.tag = element_text(size = 8))
# Figure.4

```