

Appendix 7.1 – Chapter 7 Power Calculation

Sample size was determined through an *a priori* power analysis, based on the predicted model structure for the primary analysis exploring the effect of total sleep deprivation on our overall in-game outcome measure, *goal difference* (GD). As no prior research currently exists exploring the effect of total sleep deprivation on any esports performance outcome measure, we used a published meta-analysis on the effects of total sleep deprivation on cognition (Lim & Dinges, 2010) to obtain an estimated effect size. We pooled the effect sizes provided from all cognitive domains to obtain an estimated effect size of $g = 0.435$. To obtain an estimate for pooled standard deviation, we explored a large (21,588 matches) dataset of Rocket League matches played by skill-matched individuals that has been previously described (Smithies et al., 2021), **see chapter 5**. Specifically, we took the 14 most recent matches from 23 player-opponent pairs within the dataset and used the mean of the pair standard deviations as our estimate ($SD_{\text{pooled}} = 2.799$). Using these effect size and SD estimates, we calculated the mean effect of the extended wakefulness intervention as a GD change of 1.218.

We then conducted a power analysis using a mixed-effects model (MEM) framework according to DeBruine and Barr (2021). We predicted that the model selected through our random effects selection criteria would be one with a by-pair random intercept and a by-pair random slope. To estimate the by-pair random intercept standard deviation, we took the standard deviation of pair means from the aforementioned 23 pairs as our estimate ($SD = 1.944$). To obtain an estimate of residual variance, we first calculated the standard deviation for GD in the entire previously mentioned dataset ($SD = 3.777$). We subtracted our by-pair random intercept standard deviation from this estimate to obtain our estimate of residual variance ($SD = 1.833$). We included a by-pair random slope of 1.5 to allow a non-effect of the intervention to be approximately 1SD from the mean effect. Lastly, we included a correlation between random intercept and random slope of 0.2, as per DeBruine and Barr (2021).

The R markdown for the power analysis/ simulation is provided below:

ExtWake Power Analysis

Tim D. Smithies

2023-09-07

Install the necessary packages:

```
library(lme4)

## Loading required package: Matrix
## Warning: package 'Matrix' was built under R version 4.2.3

library(lmerTest)

## Warning: package 'lmerTest' was built under R version 4.2.2
##
## Attaching package: 'lmerTest'

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

## The following object is masked from 'package:stats':
##
##      step

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.2.2
## Warning: package 'ggplot2' was built under R version 4.2.3
## Warning: package 'tibble' was built under R version 4.2.3
## Warning: package 'tidyr' was built under R version 4.2.2
## Warning: package 'readr' was built under R version 4.2.2
## Warning: package 'purrr' was built under R version 4.2.2
## Warning: package 'dplyr' was built under R version 4.2.3
## Warning: package 'stringr' was built under R version 4.2.2
## Warning: package 'lubridate' was built under R version 4.2.2

## — Attaching core tidyverse packages ————— tidyverse 2.
0.0 —
## ✓ dplyr      1.1.2      ✓ readr      2.1.4
```

```
## ✓ forcats 1.0.0      ✓ stringr 1.5.0
## ✓ ggplot2 3.4.2      ✓ tibble 3.2.1
## ✓ lubridate 1.9.2    ✓ tidyr 1.3.0
## ✓ purrr 1.0.1

## — Conflicts ————— tidyverse_conflicts() —
## ✗ tidyr::expand() masks Matrix::expand()
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag() masks stats::lag()
## ✗ tidyr::pack() masks Matrix::pack()
## ✗ tidyr::unpack() masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(dplyr)
```

Set the seed:

```
set.seed(123)
```

Set up the custom data simulation function:

```
my_sim_data <- function(
  n_subj = 19, # Number of pairs
  n_ingroup = 7, # Number of games in first session
  n_outgroup = 7, # Number of games in second session
  beta_0 = 0, # Mean GD at baseline (0)
  beta_1 = -1.21776394099438, # Estimated magnitude of ~29hr TSD effect on
  # GD
  tau_0 = 1.94444283, # Estimated by-subject random intercept standard deviation
  tau_1 = 1.5, # Estimated By-pair random slope standard deviation
  rho = 0.2, # Estimated correlation between intercept and slope
  sigma = 1.832777471) { # Estimated residual variance (standard deviation)
  items <- data.frame(
    item_id = seq_len(n_ingroup + n_outgroup),
    condition = rep(c("TSD", "CONTROL"), c(n_ingroup, n_outgroup)),
    X_i = rep(c(0, 1), c(n_ingroup, n_outgroup)))
  # variance-covariance matrix
  cov_mx <- matrix(
    c(tau_0^2, rho * tau_0 * tau_1, rho * tau_0 * tau_1, tau_1^2),
    nrow = 2, byrow = TRUE)
  subjects <- data.frame(subj_id = seq_len(n_subj),
    MASS::mvrnorm( n = n_subj, mu = c(T_0s = 0, T_1s = 0
), Sigma = cov_mx))
  crossing(subjects, items) %>% mutate(e_si = rnorm(nrow(.),
    mean = 0, sd = sigma),
    GD = beta_0 + T_0s +
      (beta_1 + T_1s) * X_i + e_si) %>%
```



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

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00293755 (tol = 0.002, component 1)

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

sims %>% filter(effect == "fixed") %>% select(term, estimate, p.value)

## # A tibble: 2,000 × 3
##   term          estimate p.value
##   <chr>          <dbl>   <dbl>
## 1 (Intercept)    -0.127 0.816
## 2 X_i           -1.69  0.00112
## 3 (Intercept)     0.245 0.631
## 4 X_i           -0.635 0.122
## 5 (Intercept)    -0.662 0.158
## 6 X_i           -1.22  0.0132
## 7 (Intercept)     1.12  0.0395
## 8 X_i           -1.19  0.0196
## 9 (Intercept)    -0.410 0.226
## 10 X_i          -1.10  0.0115
## # i 1,990 more rows
```

Calculate the mean estimates and power for specified alpha

```
alpha <- 0.05
sims %>% filter(effect == "fixed") %>% group_by(term) %>% summarize(
  mean_estimate = mean(estimate), meas_se = mean(std.error),
  power = mean(p.value < alpha), .groups = "drop")

## # A tibble: 2 × 4
##   term          mean_estimate meas_se power
##   <chr>          <dbl>   <dbl> <dbl>
## 1 (Intercept)    -0.0195   0.469 0.059
## 2 X_i           -1.21     0.406 0.805
```

The power to detect the proposed effect is presented as power for X_i

Power at 19 pairs = 0.8