

# Crossover Trial Power Using Simulation

*Eamonn O'Brien*

*23 July, 2019*

## Contents

.....	3
Function to simulate crossover trial using lme4 .....	3
Function to simulate crossover trial using nlme .....	4
Computing Environment .....	5

**Contents**

**List of Figures**

**List of Tables**

## Function to simulate crossover trial using lme4

```

# n - number of patients in each order group sdW - within
# patient standard deviation sdB - between patient standard
# deviation beta - coefficient vector c(Intercept, Treatment,
# Order, Treatment:Order)

require("lme4")

simulatec <- function(n = 10, sdW = 4, sdB = 1, beta = c(8, 1,
  0, 0), alpha = 0.05) {

  # generate data
  Patient <- as.factor(rep(1:(2 * n), rep(2, 2 * n)))
  Treatment <- c(rep(c("Treatment1", "Treatment2"), n), rep(c("Treatment2",
    "Treatment1"), n))
  Order <- rep(c("First", "Second"), 2 * n)
  Data <- data.frame(Patient, Treatment, Order)
  FMat <- model.matrix(~Treatment * Order, data = Data)
  RMat <- model.matrix(~0 + Patient, data = Data)
  Response <- FMat %>% beta + RMat %>% rnorm(2 * n, 0, sdB) +
    rnorm(4 * n, 0, sdW)
  Data$Response <- Response

  # analyse
  Fit <- lmer(Response ~ (1 | Patient) + Treatment * Order,
    data = Data)
  Est <- fixef(Fit)[2]
  Ste <- sqrt(vcov(Fit)[2, 2])
  prod(Est + c(-1, 1) * qnorm(1 - alpha/2) * Ste) > 0
}

# power for n=20 and treatment effect 4
mean(replicate(1000, simulatec(n = 20, beta = c(8, 4, 0, 0))))

[1] 0.868

# (many warning messages: boundary (singular) fit: see
# ?isSingular)

```

## Function to simulate crossover trial using nlme

```

# no warning messages for equivalent analyses compared to
# lme4 A random effect very near zero seems to be the reason
# for the warning message using lme4

require(nlme)

simulaten <- function(n = 10, sdW = 4, sdB = 1, beta = c(8, 1,
  0, 0), alpha = 0.05) {

  # generate data
  Patient <- as.factor(rep(1:(2 * n), rep(2, 2 * n)))
  Treatment <- c(rep(c("Treatment1", "Treatment2"), n), rep(c("Treatment2",
    "Treatment1"), n))
  Order <- rep(c("First", "Second"), 2 * n)
  Data <- data.frame(Patient, Treatment, Order)
  FMat <- model.matrix(~Treatment * Order, data = Data)
  RMat <- model.matrix(~0 + Patient, data = Data)
  Response <- FMat %>% beta + RMat %>% rnorm(2 * n, 0, sdB) +
    rnorm(4 * n, 0, sdW)
  Data$Response <- Response

  # analyse
  Fit <- lme(Response ~ Treatment * Order, random = ~1 | Patient,
    data = Data, na.action = "na.omit")
  Est <- fixed.effects(Fit)[2]
  Ste <- sqrt(vcov(Fit)[2, 2])
  prod(Est + c(-1, 1) * qnorm(1 - alpha/2) * Ste) > 0
}

# power for n=20 and treatment effect 4
mean(replicate(1000, simulaten(n = 20, beta = c(8, 4, 0, 0))))

```

```
[1] 0.855
```

## Computing Environment

### `sessionInfo()`

R version 3.6.1 (2019-07-05)

Platform: x86\_64-w64-mingw32/x64 (64-bit)

Running under: Windows 10 x64 (build 17134)

Matrix products: default

locale:

[1] LC\_COLLATE=English\_United Kingdom.1252

[2] LC\_CTYPE=English\_United Kingdom.1252

[3] LC\_MONETARY=English\_United Kingdom.1252

[4] LC\_NUMERIC=C

[5] LC\_TIME=English\_United Kingdom.1252

attached base packages:

[1] stats graphics grDevices utils datasets

[6] methods base

other attached packages:

[1] nlme\_3.1-140 lme4\_1.1-21 Matrix\_1.2-17 knitr\_1.23

loaded via a namespace (and not attached):

[1] Rcpp\_1.0.1 lattice\_0.20-38 digest\_0.6.20

[4] MASS\_7.3-51.4 grid\_3.6.1 formatR\_1.7

[7] magrittr\_1.5 evaluate\_0.14 stringi\_1.4.3

[10] minqa\_1.2.4 nloptr\_1.2.1 boot\_1.3-22

[13] rmarkdown\_1.14 splines\_3.6.1 tools\_3.6.1

[16] stringr\_1.4.0 xfun\_0.8 yaml\_2.2.0

[19] compiler\_3.6.1 htmltools\_0.3.6

This took 74.17 seconds to execute.