# Simple GLM to mixed model: distribution of standardized mean effect

*Han Bossier*

*15/2/2018*

## Contents

# 1 Introduction

Introduce. For final report: change number of simulations!

First note the following two factors:

$$J = \left(1 - \frac{3}{4(n-1)-1}\right) \tag{1}$$

$$h = \frac{\Gamma\left(\frac{N-1}{2}\right)}{\sqrt{\frac{N-1}{2}}\Gamma\left(\frac{N-2}{2}\right)}. \tag{2}$$

Now let us assume we have a univariate response variable $Y \sim N(\mu, \sigma)$. Furthermore denote $N$ as the sample size. Then, we have:

$$\delta = \frac{\mu}{\sigma} \tag{3}$$

$$d = \frac{\overline{Y}}{S} \tag{4}$$

$$g = d \times J \tag{5}$$

$$g^c = d \times h. \tag{6}$$

As well as:

$$\text{Var}(d) = \frac{(N-1)(1+N\delta^2)}{N(N-3)} - \frac{\delta^2}{h^2} \tag{7}$$

$$\text{Var}(g) = J^2 \times \text{Var}(d) \tag{8}$$

$$\text{Var}(g^c) = h^2 \times \text{Var}(d). \tag{9}$$

We know that:

$$\text{E}(d) = \frac{\mu}{\sigma} \times h^{-1} \tag{10}$$

$$= \delta \times h^{-1} \tag{11}$$

## 1.1 General parameters

```
# Number of simulations
nsim <- 1000

# Number of participants
```

```r
nsub <- 20

# Seed
set.seed(exp(pi) * pi)
```

## 2 Simple model without intercept

Setting.

### 2.1 Data generation

$$Y_i = \beta_1 X + \varepsilon_i, \quad i = 1, \ldots, N$$

with $\beta_1 = 3$, $X = 1$, $\varepsilon \sim N(0, \sigma^2)$ and $\sigma = 4$.

```r
# Parameters for this section
mu <- 3
sigma <- 4
delta <- mu/sigma
beta1 <- mu
X <- 1

# Empty vectors
d <- g <- gc <- varsD <- varsG <- varsGC <- vector()

# Start for loop
for(i in 1:nsim){
  # Generate N datapoints (Y)
  Y <- beta1*X + rnorm(nsub, 0, sigma)

  # Estimate standardized effects
  d_sim <- mean(Y)/sd(Y)
  d <- c(d, d_sim)
  g <- c(g,
         d_sim * corrJ(nsub))
  gc <- c(gc,
          d_sim * corrH(nsub))

  # Estimated variance
  varsD <- c(varsD,
             varD(d = d_sim, N = nsub))
  varsG <- c(varsG,
             varD(d = d_sim, N = nsub) * corrJ(N = nsub)**2)
  varsGC <- c(varsGC,
             varD(d = d_sim, N = nsub) * corrH(N = nsub)**2)

  # Reset d_sim
  rm(d_sim)
}
```

### 2.2 Monte-Carlo simulation results

```
options(scipen = 5)
expec <- data.frame('Parameter' = c('Cohen d', 'Hedges g', 'Unbiased Hedges g'),
           'Estimate' = c(mean(d), mean(g), mean(gc)),
           'SD' = c(sd(d), sd(g), sd(gc)),
           'TrueValue' = delta)
knitr::kable(expec %>% mutate(StanBias = (Estimate - TrueValue)/SD) %>%
  mutate(MSE = c(
    (var((d - delta)**2) * (nsim - 1) / nsim),
    (var((g - delta)**2) * (nsim - 1) / nsim),
    var((gc - delta)**2) * (nsim - 1) / nsim)) %>%
  select(-SD) %>%
  mutate('Avg(Var(theta))' =
           c(mean(varsD), mean(varsG), mean(varsGC))))
```

| Parameter | Estimate | TrueValue | StanBias | MSE | Avg(Var(theta)) |
|---|---|---|---|---|---|
| Cohen d | 0.7784150 | 0.75 | 0.1049369 | 0.0149693 | 0.0740933 |
| Hedges g | 0.7472784 | 0.75 | -0.0104696 | 0.0117139 | 0.0682843 |
| Unbiased Hedges g | 0.7472086 | 0.75 | -0.0107391 | 0.0117077 | 0.0682716 |

# 3 Single subject GLM with BOLD response

Generate time series for one subject, where $t$ is used to denote the scan in the time series. We now use following the following linear model in one voxel:

$$Y_t = \beta_0 + \beta_1 X + \varepsilon_t, \quad t = 1, \ldots, T.$$

Here, $X$ is a design matrix obtained by convoluting an ON/OFF blocked design with a canonical HRF. Furthermore, set $\beta_0 = 100$, $\beta_1 = 3$, $\varepsilon \sim N(0, \sigma^2)$ and $\sigma = 100$.

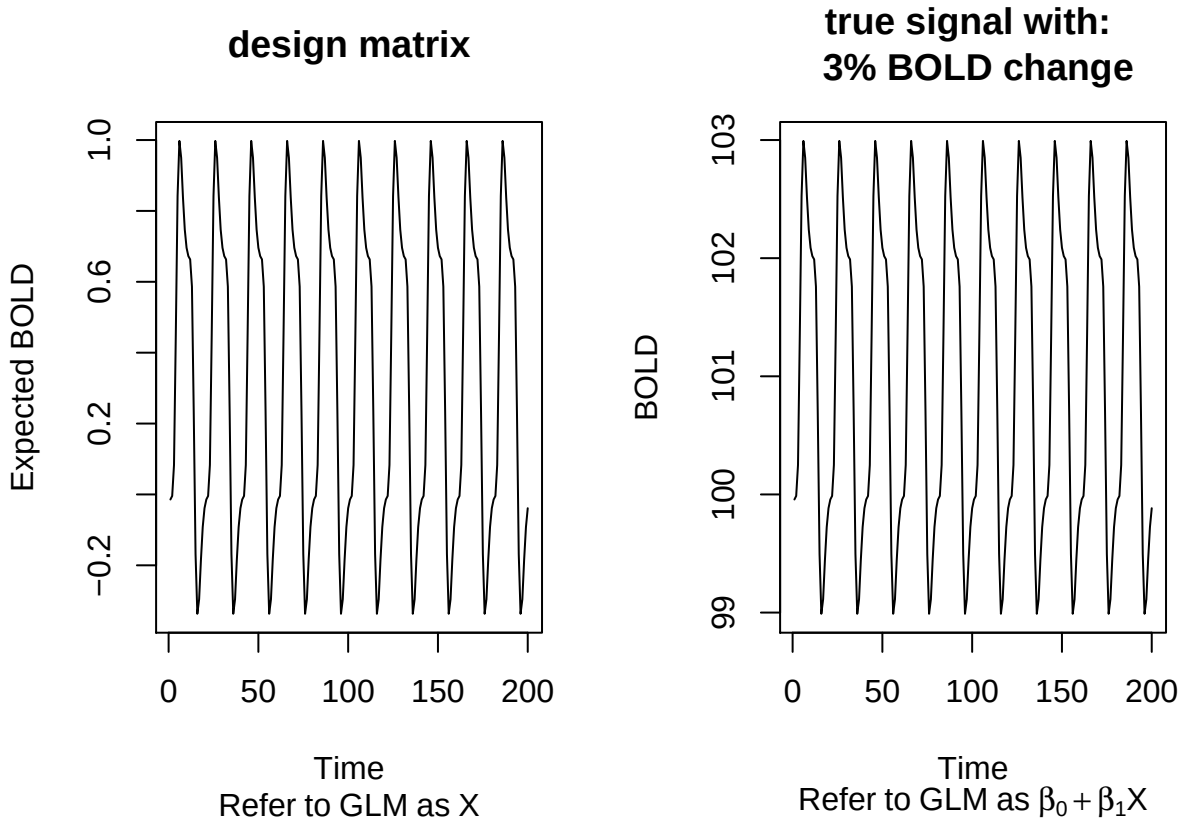
```
# Parameters for this section
mu <- 3
sigma <- 100
delta <- mu/sigma
beta0 <- 100
beta1 <- mu


# Signal characteristics
TR <- 2
nscan <- 200
total <- TR*nscan
on1 <- seq(1,total,40)
onsets <- list(on1)
duration <- list(20)

# Generating a design matrix: convolution of block design with double-gamma HRF
X <- neuRosim::simprepTemporal(total,1,onsets = onsets,
                               effectsize = 1, durations = duration,
                               TR = TR, acc = 0.1, hrf = "double-gamma")
```

```r
# X vector for one subject = predicted signal
X_s <- neuRosim::simTSfmri(design=X, base=0, SNR=1, noise="none", verbose=FALSE)

# Plot
par(mfrow = c(1,2))
plot(X_s, type = 'l', main = 'design matrix',
     sub = expression(Refer ~ to ~ GLM ~ as ~ X),
     ylab = 'Expected BOLD',
     xlab = 'Time')
plot(beta0 + beta1 * X_s, type = 'l', main =
       paste0('true signal with: \n ', beta1,'% BOLD change'),
     sub = expression(Refer ~ to ~ GLM ~ as ~ beta[0] + beta[1] * X),
    ylab = 'BOLD', xlab = 'Time')
```




### 3.1 Data generation

Let us now generate data. We estimate the standardized effect of **each** single subject through:

$$d = \frac{\hat{\beta}_1}{\sigma},$$

where $\sigma$ corresponds to the estimated residual standard error obtained by fitting the GLM. That is:

$$\hat{\sigma} = \sqrt{\sum_{t=1}^{T} \frac{(Y - \hat{Y})^2}{T - 1}}$$

We know also have:

$$g = d \times J$$

and

$$g^c = d \times h.$$

However, note that both $J$ and $h$ (and by extension the variance of the standardized effects) now depend on the number of scans (= 200) as sample size.

```
# Empty vectors
d <- g <- gc <- varsD <- varsG <- varsGC <- vector()

# Start simulation for loop
for(i in 1:nsim){
  # Generate data
  Y <- beta0 + beta1*X_s + rnorm(n = nscan, mean = 0, sd = sigma)

  # Fit GLM
  estBeta1 <- lm(Y ~ X_s)$coef['X_s']
  estSigma <- summary(lm(Y ~ X_s))$sigma

  # Estimate standardized effects
  d_sim <- estBeta1/estSigma
  d <- c(d, d_sim)
  g <- c(g,
         d_sim * corrJ(nscan))
  gc <- c(gc,
          d_sim * corrH(nscan))

  # Estimated variance
  varsD <- c(varsD,
             varD(d = d_sim, N = nscan))
  varsG <- c(varsG,
             varD(d = d_sim, N = nscan) * corrJ(N = nscan)**2)
  varsGC <- c(varsGC,
             varD(d = d_sim, N = nscan) * corrH(N = nscan)**2)

  # Reset d_sim
  rm(d_sim)
}
```

## 3.2 Monte-Carlo simulation results

Again, we look at the results. The true value is now:

$$\delta = \frac{\mu}{\sigma} = \frac{3}{100} = 0.03.$$

```
options(scipen = 5)
expec <- data.frame('Parameter' = c('Cohen d', 'Hedges g', 'Unbiased Hedges g'),
           'Estimate' = c(mean(d), mean(g), mean(gc)),
           'SD' = c(sd(d), sd(g), sd(gc)),
           'TrueValue' = delta)
knitr::kable(expec %>% mutate(StanBias = (Estimate - TrueValue)/SD) %>%
  mutate(MSE = c(
```

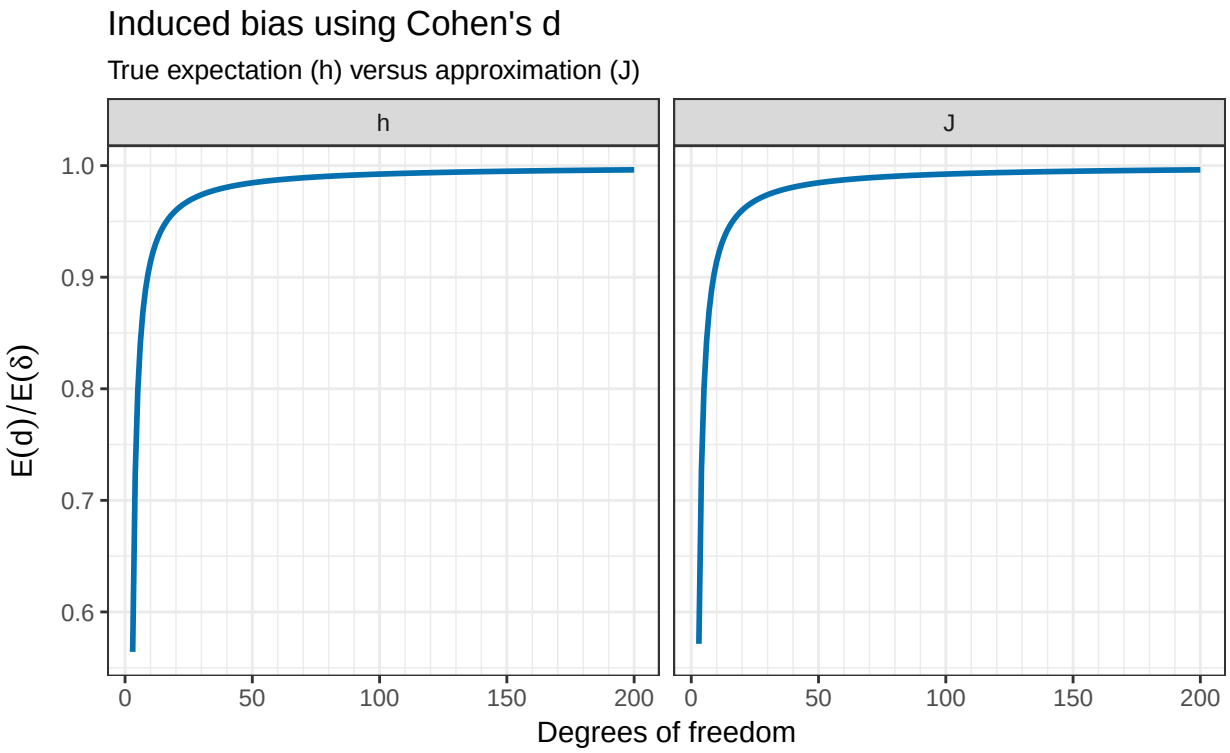
```
    (var((d - delta)**2) * (nsim - 1) / nsim),
    (var((g - delta)**2) * (nsim - 1) / nsim),
    var((gc - delta)**2) * (nsim - 1) / nsim)) %>%
  select(-SD) %>%
  mutate('Avg(Var(theta))' =
           c(mean(varsD), mean(varsG), mean(varsGC))),
  digits = 4)
```

| Parameter | Estimate | TrueValue | StanBias | MSE | Avg(Var(theta)) |
|---|---|---|---|---|---|
| Cohen d | 0.0351 | 0.03 | 0.0321 | 0.0014 | 0.0051 |
| Hedges g | 0.0350 | 0.03 | 0.0314 | 0.0014 | 0.0050 |
| Unbiased Hedges g | 0.0350 | 0.03 | 0.0314 | 0.0014 | 0.0050 |

Note that results are close to each other as number of scans is high. This is demonstrated in the following plot:

```
CorrFactor <- data.frame('df' = 3:200,
            'CorrectionF' = rep(c('J', 'h'), each = length(3:200)),
            'Value' = c(corrJ(3:200),
               corrH(3:200)))

ggplot(CorrFactor, aes(x = df, y = Value)) +
  geom_line(colour = '#0570b0', size = 1) +
  scale_x_continuous('Degrees of freedom',
                     minor_breaks = seq(10, 200, by = 10)) +
  scale_y_continuous(expression(E(d)/E(delta))) +
  ggtitle("Induced bias using Cohen's d",
          subtitle = 'True expectation (h) versus approximation (J)') +
  facet_grid(~ CorrectionF) +
  theme_bw()
```




# 4 Group study of BOLD responses

In this section, we generate time series for each subject and then combine all subjects into a group analysis. Standardized mean effects are then calculated at the group level instead of at individual subject level. We shall explore two simulation approaches. The first one is a mixed model where we generate data in one stage (within and between subjects). In the section thereafter, we will generate and analyse data using the typical two-stage approach.

## 4.1 Full mixed model approach

## 4.2 Two stage approach

The group analysis consists of fitting a GLM on the estimated first-level parameters (for each subject) using OLS. Hence we have:

$$Y_{it} = \beta_0 + \beta_1 X + \varepsilon_{it}, \quad i = 1, \ldots, N \quad \text{and} \quad t = 1, \ldots, T. \tag{12}$$

Note that we assume no autocorrelation in $\text{Var}(\varepsilon) = \sigma^2$. In the second stage, we get:

$$Y_G = \beta_1^* X_G + \varepsilon^*, \tag{13}$$

where $Y_G$ is the vector of estimated first level parameters ($\hat{\beta}_1$) and $X_G$ equals a column of 1's with length $N$. In this case, $\varepsilon^* \sim N(0, \eta^2 + \text{Var}(\widehat{\beta_1}))$. Denote $\sigma_G^2$ as $\text{Var}(\varepsilon^*)$ and note that is a mixed error component

containing both variability of the estimation at the first level and a between-subject variability component $\eta^2$. For now, we assume $\eta = 0$.

Furthermore, we have:

$$\text{Var}(\widehat{\beta}_1) = \frac{\widehat{\sigma}^2}{\sum_{t=1}^{T}(X_t - \overline{X})^2} \tag{14}$$

In matrix notation, this is:

$$\text{Var}(\boldsymbol{\beta}) = \widehat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}, \tag{15}$$

where $\mathbf{X}$ is the matrix containing both the intercept and the convoluted design. The diagonal of $\text{Var}(\boldsymbol{\beta})$ then gives the variances of the estimated parameters.

### 4.2.1 Generate data

```
# Parameters for this section
mu <- 3
sigma <- 25
beta0 <- 100
beta1 <- mu

# Between-subject variability
eta <- 1

# Empty vectors
d <- g <- gc <- varsD <- varsG <- varsGC <- vector()

# Start simulation for loop
for(i in 1:nsim){

  # Vector of estimated beta parameters
  estBeta <- vector()

  # For loop over the subjects
  for(s in 1:nsub){
    # Time series for this subject
    Y_s <- beta0 + beta1*X_s + rnorm(n = nscan, mean = 0, sd = sigma)

    # Estimated beta1 for this subject
    estBeta1_s <- lm(Y_s ~ X_s)$coef['X_s']

    # In vector
    estBeta <- c(estBeta, as.numeric(estBeta1_s))
  }

  # Add between-subject variability (eta can be zero as well)
  Yg <- estBeta + rnorm(n = nsub, mean = 0, sd = eta)
```

```
  # Fit GLM at group level
  estBetaStar1 <- lm(Yg ~ 1)$coef['(Intercept)']
  estSigmaStar <- summary(lm(Yg ~ 1))$sigma

  # Estimate standardized effects
  d_sim <- as.numeric(estBetaStar1/estSigmaStar)
  d <- c(d, d_sim)
  g <- c(g,
         d_sim * corrJ(nsub))
  gc <- c(gc,
          d_sim * corrH(nsub))

  # Estimated variance
  varsD <- c(varsD,
             varD(d = d_sim, N = nsub))
  varsG <- c(varsG,
             varD(d = d_sim, N = nsub) * corrJ(N = nsub)**2)
  varsGC <- c(varsGC,
             varD(d = d_sim, N = nsub) * corrH(N = nsub)**2)

  # Reset
  rm(d_sim, Yg, estBetaStar1, estSigmaStar)
}
```

#### 4.2.1.1 Note on data generation

To induce between-subject variability (when $\eta > 0$), we can either generate a specific $\beta_{1i}$ for each subject $i$:

```
for(s in 1:nsub){
  # Random beta1 for this subject
  beta1S <- rnorm(n = 1, mean = beta1, sd = 0)

  # Time series for this subject
  Y_s <- beta0 + beta1S*X_s + rnorm(n = nscan, mean = 0, sd = sigma)

  # Estimated beta1 for this subject
  estBeta1_s <- lm(Y_s ~ X_s)$coef['X_s']

  # In vector
  estBeta <- c(estBeta, as.numeric(estBeta1_s))
}
Yg <- estBeta
```

Or we could take the vector of first-level responses ($\widehat{\beta}_{1i}$) and induce variability here:

```
for(s in 1:nsub){
  # Time series for this subject
  Y_s <- beta0 + beta1*X_s + rnorm(n = nscan, mean = 0, sd = sigma)

  # Estimated beta1 for this subject
  estBeta1_s <- lm(Y_s ~ X_s)$coef['X_s']

  # In vector
  estBeta <- c(estBeta, as.numeric(estBeta1_s))
```

```
}
# Add between-subject variability (eta can be zero as well)
Yg <- estBeta + rnorm(n = nsub, mean = 0, sd = eta)
```

The first approach suggests a random slope approach, while the second matches more closely the two-stage fMRI notation.

### 4.2.2 Monte-Carlo simulation results

We now have the true standardized mean effect at the group level as:

$$\delta = \frac{\mu}{\sigma_G^*} \tag{16}$$

$$= \frac{\beta_1^*}{\sigma_G^*} \tag{17}$$

$$= \frac{\beta_1^*}{\sqrt{\eta^2 + \sigma^2 (X'X)^{-1}}} \tag{18}$$

Note that $\delta$ defined at the group level now depends on the design matrix of the first level. Furthermore note that the second element of the diagonal on $(\mathbf{X}'\mathbf{X})^{-1}$ equals:

$$\text{diag}(\mathbf{X}'\mathbf{X})_2^{-1} = \frac{1}{\sum_{t=1}^{T}(X_t - \overline{X})^2}$$

as is demonstrated:

```
# Extend the design matrix with the intercept
xIN <- cbind(1,X_s)
diag(solve(t(xIN)%*%xIN))[2]
```

```
       X_s
0.02543824
```

```
1/(var(X_s) * (nscan - 1))
```

```
[1] 0.02543824
```

Let us now look at the Monte-Carlo simulation results.

```
# True values
varBeta1 <- sigma^2 * diag(solve(t(xIN)%*%xIN))[2]
delta <- mu/(sqrt(eta + varBeta1))

expec <- data.frame('Parameter' = c('Cohen d', 'Hedges g', 'Unbiased Hedges g'),
           'Estimate' = c(mean(d), mean(g), mean(gc)),
           'SD' = c(sd(d), sd(g), sd(gc)),
           'TrueValue' = as.numeric(delta))
knitr::kable(expec %>% mutate(StanBias = (Estimate - TrueValue)/SD) %>%
  mutate(MSE = c(
    (var((d - delta)**2) * (nsim - 1) / nsim),
    (var((g - delta)**2) * (nsim - 1) / nsim),
    var((gc - delta)**2) * (nsim - 1) / nsim)) %>%
  select(-SD) %>%
  mutate('Avg(Var(theta))' =
```

```
          c(mean(varsD), mean(varsG), mean(varsGC))),
digits = 4)
```

| Parameter | Estimate | TrueValue | StanBias | MSE | Avg(Var(theta)) |
|---|---|---|---|---|---|
| Cohen d | 0.7552 | 0.7298 | 0.0924 | 0.0177 | 0.0559 |
| Hedges g | 0.7250 | 0.7298 | -0.0182 | 0.0136 | 0.0515 |
| Unbiased Hedges g | 0.7249 | 0.7298 | -0.0185 | 0.0136 | 0.0515 |