A few weeks ago, I was at the annual meeting of the [NIH Collaboratory](https://rethinkingclinicaltrials.org/), which is an innovative collection of collaboratory cores, demonstration projects, and NIH Institutes and Centers that is developing new models for implementing and supporting large-scale health services research. A study I am involved with – *Primary Palliative Care for Emergency Medicine* – is one of the demonstration projects in this collaboratory.

The second day of this meeting included four panels devoted to the design and analysis of embedded pragmatic clinical trials, and focused on the challenges of conducting rigorous research in the real-world context of a health delivery system. The keynote address that started off the day was presented by David Murray of NIH, who talked about the challenges and limitations of cluster randomized trials. (I’ve written before on issues related to clustered randomized trials, including [here](https://www.rdatagen.net/post/what-matters-more-in-a-cluster-randomized-trial-number-or-size/).)

In particular, Dr. Murray talked a great deal about stepped-wedge designs, which have become a quite popular tool in health services research. (I described stepped-wedge designs [here](https://www.rdatagen.net/post/alternatives-to-stepped-wedge-designs/).) A big takeaway from the talk was that we must be cognizant of the underlying assumptions of the models used to estimate treatment effects; being unaware can lead to biased estimates of treatment effects, or more likely, biased estimates of uncertainty.

**Intra-cluster correlations**

If outcomes of subjects in a study are correlated in any way (e.g. they received care from the same health care provider), we do not learn as much information from each individual study participant as we would in the case where there is no correlation. In a parallel designed cluster randomized trial (where half of the clusters receive an intervention and the other half do not), we expect that the outcomes will be correlated *within* each cluster, though not *across* clusters. (This is not true if the clusters are themselves clustered, in which case we would have a 2-level clustered study.) This intra-cluster correlation (ICC) increases sample size requirements and reduces precision/power.

A common way to model correlation explicitly in a cluster randomized trial is to conceive of a random effects model like this:

\[(1) \qquad \qquad Y\_{ic} = \mu + \beta\_1X\_{c} + b\_c + e\_{ic},\]

where \(Y\_{ic}\) is a continuous outcome for subject \(i\) in cluster \(c\), and \(X\_c\) is a treatment indicator for cluster \(c\) (either 0 or 1). The underlying structural parameters are \(\mu\), the grand mean, and \(\beta\_1\), the treatment effect. The unobserved random effects are, \(b\_c \sim N(0, \sigma^2\_b)\), the normally distributed group level effect, and \(e\_{ic} \sim N(0, \sigma^2\_e)\), the normally distributed individual-level effect. (This is often referred to as the “error” term, but that doesn’t adequately describe what is really unmeasured individual variation.)

The correlation between any two subjects \(i\) and \(j\) in the *same* cluster \(c\) is:

\[ cor(Y\_{ic}, Y\_{jc}) = \frac{cov(Y\_{ic}, Y\_{jc})} {\sqrt {var(Y\_{ic})var(Y\_{jc})}} \]

\(cov(Y\_{ic}, Y\_{jc})\) can be written in terms of the parameters in the underlying data generating process:

\[  
\begin{aligned}  
cov(Y\_{ic}, Y\_{jc}) &= cov(\mu + \beta\_1X\_c + b\_c + e\_{ic}, \mu + \beta\_1X\_c + b\_c + e\_{jc}) \\  
&=cov(b\_c, b\_c) + cov(e\_{ic},e\_{jc} ) \\  
&=\sigma^2\_b + 0 \\  
&=\sigma^2\_b  
\end{aligned}  
\]

The terms simplify since the cluster level effects are independent of the individual level effects (and all the fixed effects in the model) and the individual level effects are independent of each other. The within-period intra-cluster co-variance depends only on the between cluster variation.

The total variance of the outcomes \(Y\_{ic}\) is:

\[  
\begin{aligned}  
var(Y\_{ic}) &= var(\mu + \beta\_1X\_c + b\_c + e\_{ic}) \\  
&= var(b\_c) + var(e\_{ic}) \\  
&= \sigma^2\_b + \sigma^2\_e  
\end{aligned}  
\]

Substituting all of this into the original equation gives us the intra-cluster correlation for any two subjects in the cluster:

\[  
\begin{aligned}  
cor(Y\_{ic}, Y\_{jc}) &= \frac{cov(Y\_{ic}, Y\_{jc})} {\sqrt {var(Y\_{ic})var(Y\_{jc})}} \\  
\\  
ICC &= \frac{\sigma^2\_b}{\sigma^2\_b + \sigma^2\_e}  
\end{aligned}  
\]

So, the correlation between any two subjects in a cluster increases as the variation *between* clusters increases.

**Cluster randomization when time matters**

Moving beyond the parallel design to the stepped-wedge design, time starts to play a very important role. It is important to ensure that we do not confound treatment and time effects; we have to be careful that we do not attribute the general changes over time to the intervention. This is accomplished by introducing a time trend into the model. (Actually, it seems more common to include a time-specific effect so that each time period has its own effect. However, for simulation purposes, I will will assume a linear trend.)

In the stepped-wedge design, we are essentially estimating within-cluster treatment effects by comparing the cluster with itself pre- and post-intervention. To estimate sample size and precision (or power), it is no longer sufficient to consider a single ICC, because there are now multiple ICC’s – the within-period ICC and the between-period ICC’s. The within-period ICC is what we defined in the parallel design (since we effectively treated all observations as occurring in the same period.) Now we also need to consider the expected correlation of two individuals in the *same* cluster in *different* time periods.

If we do not properly account for within-period ICC and the between-period ICC’s in either the planning or analysis stages, we run the risk of generating biased estimates.

My primary aim is to describe possible data generating processes for the stepped wedge design and what implications they have for both the within-period and between-period ICC’s. I will generate data to confirm that observed ICC’s match up well with the theoretical expectations. This week I will consider the simplest model, one that is frequently used but whose assumptions may not be realistic in many applications. In a follow-up post, I will consider more flexible data generating processes.

**Constant ICC’s over time**

Here is probably the simplest model that can be conceived for a process underlying the stepped-wedge design:

\[  
(2) \qquad \qquad Y\_{ict} = \mu + \beta\_0t + \beta\_1X\_{ct} + b\_c + e\_{ict}  
\]

As before, the unobserved random effects are \(b\_c \sim N(0, \sigma^2\_b)\) and \(e\_{ict} \sim N(0, \sigma^2\_e)\). The key differences between this model compared to the parallel design is the time trend and time-dependent treatment indicator. The time trend accounts for the fact that the outcome may change over time regardless of the intervention. And since the cluster will be in both the control and intervention states we need to have an time-dependent intervention indicator. (This model is a slight variation on the *Hussey and Hughes* model, which includes a time-specific effect \(\beta\_t\) rather than a linear time trend. This paper by [*Kasza et al*](https://journals.sagepub.com/doi/full/10.1177/0962280217734981) describes this stepped-wedge model, and several others, in much greater detail.)

The *within-period* ICC from this is model is:

\[  
\begin{aligned}  
cor(Y\_{ict}, Y\_{jct}) &= cor(\mu + \beta\_0t + \beta\_1X\_{ct} + b\_c + e\_{ict}, \ \mu + \beta\_0t + \beta\_1X\_{ct} + b\_c + e\_{jct}) \\  
\\  
ICC\_{tt}&= \frac{\sigma^2\_b}{\sigma^2\_b + \sigma^2\_e}  
\end{aligned}  
\]

I have omitted the intermediary steps, but the logic is the same as in the parallel design case. The within-period ICC under this model is also the same as the ICC in the parallel design.

More importantly, in this case the *between-period* ICC turns out to be the same as the *within-period* ICC. For the *between-period* ICC, we are estimating the expected correlation between any two subjects \(i\) and \(j\) in cluster \(c\), one in time period \(t\) and the other in time period \(t^\prime\) \((t \ne t^\prime)\):

\[  
\begin{aligned}  
cor(Y\_{ict}, Y\_{jct^\prime}) &= cor(\mu + \beta\_0t + \beta\_1X\_{ct} + b\_c + e\_{ict}, \ \mu + \beta\_0t^\prime + \beta\_1X\_{ct^\prime} + b\_c + e\_{jct^\prime}) \\  
\\  
ICC\_{tt^\prime}&= \frac{\sigma^2\_b}{\sigma^2\_b + \sigma^2\_e}  
\end{aligned}  
\]

Under this seemingly reasonable (and popular) model, we are making a big assumption that the within-period ICC and between-period ICC’s are equal and constant throughout the study. This may or may not be reasonable – but it is important to acknowledge the assumption and to make sure we justify that choice.

**Generating data to simulate a stepped-wedge design**

I’ve generated data from a stepped-wedge design [before](https://www.rdatagen.net/post/simstudy-update-stepped-wedge-treatment-assignment/) on this blog, but will repeat the details here. For the data definitions, we define the variance of the cluster-specific effects, the cluster sizes, and the outcome model.

defc <- defData(varname = "ceffect", formula = 0, variance = 0.15,

dist = "normal", id = "cluster")

defc <- defData(defc, "m", formula = 10, dist = "nonrandom")

defa <- defDataAdd(varname = "Y",

formula = "0 + 0.10 \* period + 1 \* rx + ceffect",

variance = 2, dist = "normal")

The data generation follows this sequence: cluster data, temporal data, stepped-wedge treatment assignment, and individual (within cluster) data:

dc <- genData(100, defc)

dp <- addPeriods(dc, 7, "cluster")

dp <- trtStepWedge(dp, "cluster", nWaves = 4, lenWaves = 1, startPer = 2)

dd <- genCluster(dp, cLevelVar = "timeID", "m", "id")

dd <- addColumns(defa, dd)

dd

## cluster period ceffect m timeID startTrt rx id Y

## 1: 1 0 -0.073 10 1 2 0 1 -2.12

## 2: 1 0 -0.073 10 1 2 0 2 -1.79

## 3: 1 0 -0.073 10 1 2 0 3 1.53

## 4: 1 0 -0.073 10 1 2 0 4 -1.44

## 5: 1 0 -0.073 10 1 2 0 5 2.25

## ---

## 6996: 100 6 0.414 10 700 5 1 6996 1.28

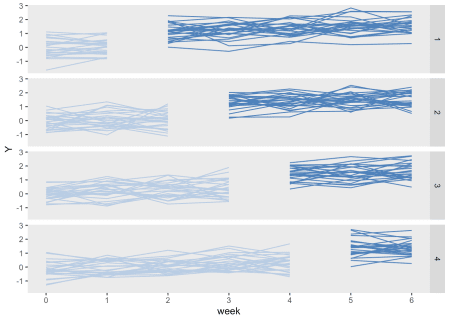
## 6997: 100 6 0.414 10 700 5 1 6997 0.30

## 6998: 100 6 0.414 10 700 5 1 6998 0.94

## 6999: 100 6 0.414 10 700 5 1 6999 1.43

## 7000: 100 6 0.414 10 700 5 1 7000 0.58

It is always useful (and important) to visualize the data (regardless of whether they are simulated or real). This is the summarized cluster-level data. The clusters are grouped together in waves defined by starting point. In this case, there are 25 clusters per wave. The light blue represents pre-intervention periods, and the dark blue represents intervention periods.



**Estimating the between-period within-cluster correlation**

I want to estimate the observed between-period within cluster correlation without imposing any pre-conceived structure. In particular, I want to see if the data generated by the process defined in equation (2) above does indeed lead to constant within- and between-period ICC’s. In a future post, I will estimate the ICC using a model, but for now, I’d prefer to estimate the ICC’s directly from the data.

A 1982 paper by [*Bernard Rosner*](https://academic.oup.com/aje/article/116/4/722/52694) provides a non-parametric estimate of the *between-period* ICC. He gives this set of equations to find the correlation coefficient \(\rho\_{tt^\prime}\) for two time periods \(t\) and \(t^\prime\). In the equations, \(m\_{ct}\) represents the cluster size for cluster \(c\) in time period \(t\), and \(K\) represents the number of clusters:

\[  
\rho\_{tt^\prime} = \frac{\sum\_{c=1}^K \sum\_{i=1}^{m\_{ct}} \sum\_{j=1}^{m\_{ct^\prime}} (Y\_{ict}-\mu\_t)(Y\_{jct^\prime}-\mu\_{t^\prime})} {\left[ \left ( \sum\_{c=1}^K m\_{ct^\prime} \sum\_{i=1}^{m\_{ct}} (Y\_{ict}-\mu\_t)^2 \right ) \left ( \sum\_{c=1}^K m\_{ct} \sum\_{j=1}^{m\_{ct^\prime}} (Y\_{jct^\prime}-\mu\_{t^\prime})^2 \right )\right] ^ \frac {1}{2}}  
\]

\[  
\mu\_t = \frac{\sum\_{c=1}^K m\_{ct} m\_{ct^\prime} \mu\_{ct}}{\sum\_{c=1}^K m\_{ct} m\_{ct^\prime}} \ \ , \ \ \mu\_{t^\prime} = \frac{\sum\_{c=1}^K m\_{ct} m\_{ct^\prime} \mu\_{ct^\prime}}{\sum\_{c=1}^K m\_{ct} m\_{ct^\prime}}  
\]

\[  
\mu\_{ct} = \frac{\sum\_{i=1}^{m\_{ct}} Y\_{ict}}{m\_{ct}} \ \ , \ \ \mu\_{ct^\prime} = \frac{\sum\_{j=1}^{m\_{ct^\prime}} Y\_{jct^\prime}}{m\_{ct^\prime}}  
\]

I’ve implemented the algorithm in R, and the code is included in the addendum. One issue that came up is that as the intervention is phased in over time, the treatment effect is present for each at different times. The algorithm breaks down as a result. However, the between-period ICC can be calculated for each wave, and then we can average across the four waves.

The *within-period* ICC is estimated using a linear mixed effects model applied to each period separately, so that we estimate period-specific within-period ICC’s. The expected (constant) ICC is \(0.07 = \left(\frac{0.15}{0.15 + 2}\right)\).

The function iccs (shown below in the addendum) returns both the estimated *within-* and *between-cluster* ICC’s for a single data set. Here is the within-period ICC for the first period (actually period 0) and the between-period ICC’s using period 0:

set.seed(47463)

iccs(dd, byWave = T)[,c(22, 0:6)]

## wp0 bp01 bp02 bp03 bp04 bp05 bp06

## 1: 0.041 0.068 0.073 0.08 0.067 0.054 0.053

ICC estimates are quite variable and we can’t tell anything about the distribution from any single data set. Generating multiple replications lets us see if the estimates are close, on average, to our assumption of constant ICC’s. Here is a function to generate a single data set:

genDD <- function(defc, defa, nclust, nperiods, waves, len, start) {

dc <- genData(nclust, defc)

dp <- addPeriods(dc, nperiods, "cluster")

dp <- trtStepWedge(dp, "cluster", nWaves = waves,

lenWaves = len, startPer = start)

dd <- genCluster(dp, cLevelVar = "timeID", "m", "id")

dd <- addColumns(defa, dd)

return(dd[])

}

And here is a function to estimate 200 sets of ICC’s for 200 data sets:

icc <- mclapply(1:200,

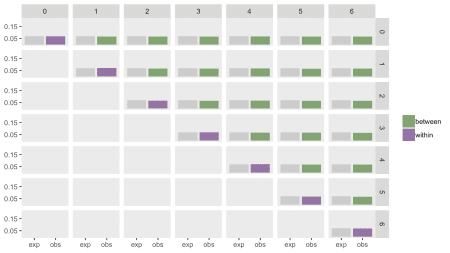
function(x) iccs(genDD(defc, defa, 100, 7, 4, 1, 2), byWave = T),

mc.cores = 4

)

observed <- sapply(rbindlist(icc), function(x) mean(x))

Averages of all the *within-* and *between-period* ICC’s were in fact quite close to the “true” value of 0.07 based on a relatively small number of replications. The plot shows the observed averages along side the expected value (shown in gray) for each of the periods generated in the data. There is little variation across both the *within-* and *between-period* ICC’s.



I’ll give you a little time to absorb this. Next time, I will consider alternative data generating processes where the the ICC’s are not necessarily constant.

References:

Kasza, J., K. Hemming, R. Hooper, J. N. S. Matthews, and A. B. Forbes. “Impact of non-uniform correlation structure on sample size and power in multiple-period cluster randomised trials.” *Statistical methods in medical research* (2017): 0962280217734981.

Rosner, Bernard. “On the estimation and testing of inter-class correlations: the general case of multiple replicates for each variable.” *American journal of epidemiology* 116, no. 4 (1982): 722-730.

**Addendum: R code for simulations**

library(lme4)

library(parallel)

Covar <- function(dx, clust, period1, period2, x\_0, x\_1) {

v0 <- dx[ctemp == clust & period == period1, Y - x\_0]

v1 <- dx[ctemp == clust & period == period2, Y - x\_1]

sum(v0 %\*% t(v1))

}

calcBP <- function(dx, period1, period2) {

# dx <- copy(d2)

# create cluster numbers starting from 1

tt <- dx[, .N, keyby = cluster]

nclust <- nrow(tt)

dx[, ctemp := rep(1:nclust, times = tt$N)]

dx <- dx[period %in% c(period1, period2)]

## Grand means

dg <- dx[, .(m=.N, mu = mean(Y)), keyby = .(ctemp, period)]

dg <- dcast(dg, formula = ctemp ~ period, value.var = c("m","mu"))

setnames(dg, c("ctemp", "m\_0", "m\_1", "mu\_0", "mu\_1"))

x\_0 <- dg[, sum(m\_0 \* m\_1 \* mu\_0)/sum(m\_0 \* m\_1)]

x\_1 <- dg[, sum(m\_0 \* m\_1 \* mu\_1)/sum(m\_0 \* m\_1)]

## Variance (denominator)

dss\_0 <- dx[period == period1, .(ss\_0 = sum((Y - x\_0)^2)),

keyby = ctemp]

dss\_0[, m\_1 := dg[, m\_1]]

v\_0 <- dss\_0[, sum(m\_1 \* ss\_0)]

dss\_1 <- dx[period == period2, .(ss\_1 = sum((Y - x\_1)^2)),

keyby = ctemp]

dss\_1[, m\_0 := dg[, m\_0]]

v\_1 <- dss\_1[, sum(m\_0 \* ss\_1)]

## Covariance

v0v1 <- sapply(1:nclust,

function(x) Covar(dx, x, period1, period2, x\_0, x\_1))

bp.icc <- sum(v0v1)/sqrt(v\_0 \* v\_1)

bp.icc

}

btwnPerICC <- function(dd, period1, period2, byWave = FALSE) {

if (byWave) {

waves <- dd[, unique(startTrt)]

bpICCs <- sapply(waves, function(x)

calcBP(dd[startTrt==x], period1, period2))

return(mean(bpICCs))

} else {

calcBP(dd, period1, period2)

}

}

withinPerICC <- function(dx) {

lmerfit <- lmer(Y~rx + (1|cluster), data = dx)

vars <- as.data.table(VarCorr(lmerfit))[, vcov]

vars[1]/sum(vars)

}

genPairs <- function(n) {

x <- combn(x = c(1:n-1), 2)

lapply(seq\_len(ncol(x)), function(i) x[,i])

}

iccs <- function(dd, byWave = FALSE) {

nperiods <- dd[, length(unique(period))]

bperiods <- genPairs(nperiods)

names <-

unlist(lapply(bperiods, function(x) paste0("bp", x[1], x[2])))

bp.icc <- sapply(bperiods,

function(x) btwnPerICC(dd, x[1], x[2], byWave))

system(paste("echo ."))

bdd.per <- lapply(1:nperiods - 1, function(x) dd[period == x])

wp.icc <- lapply(bdd.per,

function(x) withinPerICC(x))

wp.icc <- unlist(wp.icc)

nameswp <- sapply(1:nperiods - 1, function(x) paste0("wp", x))

do <- data.table(t(c(bp.icc, wp.icc)))

setnames(do, c(names, nameswp))

return(do[])

}