

R code to re-create a simulation for the nested exchangeable structure for equal cluster sizes

```
# Packages and source programs
library(geepack)
source('FUNCTIONS.R')

# Design
I <- 20      # number of clusters
n <- 30      # number of participants per cluster per period
T <- 5       # number of time periods
q <- I/(T-1) # number of clusters randomized at each step
sig <- 0.05  # significance level
p0 <- 0.3    # baseline prevalence under control
beta <- c(log(p0/(1-p0)), -0.87, -0.89, -0.91, -0.93) # gently decreasing time-effect
theta <- 0.4592 # intervention effect on the log odds ratio scale
rho1 <- 0.06  # within-period correlation
rho2 <- 0.04  # inter-period correlation
R <- (1-rho1)*diag(n*T) +
  (rho1-rho2)* kronecker(diag(T), matrix(1, nrow=n, ncol=n)) +
  rho2*matrix(1, nrow=n*T, ncol=n*T) # nested exchangeable correlation
trtSeq <- matrix(0, T-1, T)
trtSeq[upper.tri(trtSeq)] <- 1 # treatment sequences

# Simulate data
set.seed(20210513)
B <- create.B(I, T, q, beta, theta, trtSeq, n, R) # Create b's as in Qaqish equation 3
y <- create.response(I, T, q, beta, theta, trtSeq, n, B) # Simulate binary outcomes (Qaqish equation 3)

# Fit the model with the true correlation structure
# user defined correlation structure
gh <- NULL
for (i in 2:(n*T)){
  fh <- cbind(rep(i-1, length(i:(n*T))), i:(n*T))
  gh <- rbind(gh, fh)
}
cor <- matrix(NA, nrow=n*T*(n*T-1)/2, ncol=2)
# Same cluster, same time-period (rho1)
cor[,1] <- (gh[,1]<=n & gh[,2]<=n) |
  (gh[,1]>n & gh[,2]>n & gh[,1]<=2*n & gh[,2]<=2*n) |
  (gh[,1]>2*n & gh[,2]>2*n & gh[,1]<=3*n & gh[,2]<=3*n) |
  (gh[,1]>3*n & gh[,2]>3*n & gh[,1]<=4*n & gh[,2]<=4*n) |
  (gh[,1]>4*n & gh[,2]>4*n & gh[,1]<=5*n & gh[,2]<=5*n)
# Same cluster, different time-period (rho2)
cor[,2] <- 1-cor[,1]
# Replicate for all clusters
cor_p <- matrix(rep(t(cor), I), ncol=ncol(cor), byrow=TRUE)
cluster <- rep(1:I, each=T*n)
period <- rep(rep(1:T, each=n), I)
X.ij <- kronecker(trtSeq, rep(1, q))
treatment <- rep(c(t(X.ij)), each=n)
simdata_bin <- data.frame(cbind(y, cluster, period, treatment))
fit_pack <- geeglm(y~treatment+factor(period), id=cluster, data=simdata_bin,
  family=binomial(link = "logit"), scale.fix=TRUE, corstr="userdefined", zcor=cor_p)
```

```
summary(fit_pack) # mean model, sandwich SEs and correlations
```

```
##
## Call:
## geeglm(formula = y ~ treatment + factor(period), family = binomial(link = "logit"),
## data = simdata_bin, id = cluster, zcor = cor_p, corstr = "userdefined",
## scale.fix = TRUE)
##
## Coefficients:
## Estimate Std.err Wald Pr(>|W|)
## (Intercept) -0.7689 0.1484 26.861 2.19e-07 ***
## treatment 0.5979 0.1722 12.055 0.000516 ***
## factor(period)2 -0.3356 0.1611 4.337 0.037286 *
## factor(period)3 -0.3777 0.1840 4.216 0.040033 *
## factor(period)4 -0.3130 0.1821 2.955 0.085603 .
## factor(period)5 -0.5825 0.2052 8.062 0.004520 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = userdefined
## Scale is fixed.
##
## Link = identity
##
## Estimated Correlation Parameters:
## Estimate Std.err
## alpha:1 0.05822 0.01516
## alpha:2 0.04175 0.01392
## Number of clusters: 20 Maximum cluster size: 150
```

```
sqrt(fit_pack$geese$vbeta.naiv[2,2]) # model-based SE for the treatment effect
```

```
## [1] 0.1656
```

```
# Get Mancl DeRouen sandwich variance
```

```
alpha1 <- summary(fit_pack)$corr[1,1]
alpha2 <- summary(fit_pack)$corr[2,1]
var <- (1-alpha1)*diag(n*T) + (alpha1-alpha2)* kronecker(diag(T),matrix(1,nrow=n,ncol=n)) +
alpha2*matrix(1,nrow=n*T,ncol=n*T)
md_var <- md.var(fit_pack,simdata_bin,var)
sqrt(md_var) # Mancl DeRouen sandwich SE for the treatment effect
```

```
## [1] 0.1916
```

```
# Fit the model with the working exchangeable correlation structure
```

```
fit_exch <- geeglm(y~treatment+factor(period),id=cluster,data=simdata_bin,
family=binomial(link = "logit"), scale.fix=TRUE, corstr="exchangeable")
summary(fit_exch) # mean model, sandwich SEs and correlation
```

```
##
## Call:
## geeglm(formula = y ~ treatment + factor(period), family = binomial(link = "logit"),
## data = simdata_bin, id = cluster, corstr = "exchangeable",
## scale.fix = TRUE)
##
## Coefficients:
## Estimate Std.err Wald Pr(>|W|)
## (Intercept) -0.769 0.148 26.86 2.2e-07 ***
## treatment 0.612 0.173 12.49 0.00041 ***
## factor(period)2 -0.339 0.161 4.45 0.03481 *
## factor(period)3 -0.385 0.184 4.38 0.03639 *
## factor(period)4 -0.324 0.183 3.13 0.07675 .
```

```

## factor(period)5    -0.596    0.205    8.45    0.00365 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = exchangeable
## Scale is fixed.
##
## Link = identity
##
## Estimated Correlation Parameters:
##      Estimate Std.err
## alpha    0.0451  0.0139
## Number of clusters:    20 Maximum cluster size: 150
sqrt(fit_exch$geese$vbeta.naiv[2,2]) # model-based SE for the treatment effect

## [1] 0.138
# Get Mancl DeRouen sandwich variance
alpha <- summary(fit_exch)$corr[1,1]
var <- (1-alpha)*diag(n*T) + alpha*matrix(1,nrow=n*T,ncol=n*T)
md_var <- md.var(fit_exch,simdata_bin,var)
sqrt(md_var) # Mancl DeRouen sandwich SE for the treatment effect

## [1] 0.193
# Fit the model with the working independence correlation structure
fit_indep <- geeglm(y~treatment+factor(period),id=cluster,data=simdata_bin,
                    family=binomial(link = "logit"), scale.fix=TRUE, corstr="independence")
summary(fit_indep) # mean model and sandwich SEs

##
## Call:
## geeglm(formula = y ~ treatment + factor(period), family = binomial(link = "logit"),
## data = simdata_bin, id = cluster, corstr = "independence",
## scale.fix = TRUE)
##
## Coefficients:
##      Estimate Std.err Wald Pr(>|W|)
## (Intercept)    -0.769   0.148 26.88  2.2e-07 ***
## treatment        0.481   0.210  5.22   0.022 *
## factor(period)2  -0.305   0.169  3.24   0.072 .
## factor(period)3  -0.314   0.195  2.59   0.108
## factor(period)4  -0.225   0.202  1.24   0.265
## factor(period)5  -0.465   0.244  3.65   0.056 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Scale is fixed.
##
## Number of clusters:    20 Maximum cluster size: 150
sqrt(fit_indep$geese$vbeta.naiv[2,2]) # model-based SE for the treatment effect

## [1] 0.113
# Get Mancl DeRouen sandwich variance
var <- diag(n*T)
md_var <- md.var(fit_indep,simdata_bin,var)
sqrt(md_var) # Mancl DeRouen sandwich SE for the treatment effect

## [1] 0.237

```