

R code to re-create a simulation for the heterogeneous treatment structure with 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 <- 0.05  # inter-cluster correlation (control, control)
rho2 <- 0.04  # inter-cluster correlation (control, treatment)
rho3 <- 0.074 # inter-cluster correlation (treatment, treatment)
trtSeq <- matrix(0, T-1, T)
trtSeq[upper.tri(trtSeq)] <- 1 # treatment sequences

# Simulate data
set.seed(20210513)
B <- create.B.het(I, T, q, beta, theta, trtSeq, n, rho1, rho2, rho3) # 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_p <- NULL
for (t in 2:T) {
  cor <- matrix(NA, nrow=n*T*(n*T-1)/2, ncol=3)
  cor[,1] <- gh[,2] <= (t-1)*n
  cor[,2] <- gh[,1] <= (t-1)*n & gh[,2] > (t-1)*n
  cor[,3] <- gh[,1] > (t-1)*n
  cori <- matrix(rep(t(cor), q), ncol=ncol(cor), byrow=TRUE)
  cor_p <- rbind(cor_p, cori)
}
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)    -0.7999  0.1328 36.286  1.7e-09 ***
## treatment         0.5228  0.1515 11.901  0.000561 ***
## factor(period)2  -0.3234  0.1312  6.079  0.013681 *
## factor(period)3  -0.2830  0.1397  4.103  0.042796 *
## factor(period)4  -0.1745  0.1510  1.335  0.247964
## factor(period)5  -0.2338  0.1745  1.793  0.180504
## ---
## 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  0.04186 0.01683
## alpha:2  0.05471 0.02101
## alpha:3  0.13099 0.03401
## Number of clusters:  20 Maximum cluster size: 150
sqrt(fit_pack$geese$vbeta.naiv[2,2]) # model-based SE for the treatment effect

## [1] 0.1766
# Get Mancl DeRouen sandwich variance
alpha1 <- summary(fit_pack)$corr[1,1]
alpha2 <- summary(fit_pack)$corr[2,1]
alpha3 <- summary(fit_pack)$corr[3,1]
md_var <- md.var.het(fit_pack,simdata_bin,alpha1,alpha2,alpha3,n,T,q,trtSeq)
sqrt(md_var) # Mancl DeRouen sandwich SE for the treatment effect

## [1] 0.1665
# 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)    -0.799  0.133 36.21  1.8e-09 ***
## treatment         0.531  0.154 11.97  0.00054 ***
## factor(period)2  -0.274  0.135  4.13  0.04223 *
## factor(period)3  -0.266  0.163  2.66  0.10315
## factor(period)4  -0.165  0.155  1.14  0.28622
## factor(period)5  -0.242  0.175  1.92  0.16618
## ---
## 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 0.0732 0.0186
## Number of clusters: 20 Maximum cluster size: 150

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

## [1] 0.136
# 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] 0.17
# 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) -0.800 0.133 36.34 1.7e-09 ***
## treatment 0.570 0.248 5.31 0.021 *
## factor(period)2 -0.283 0.138 4.20 0.040 *
## factor(period)3 -0.285 0.161 3.12 0.077 .
## factor(period)4 -0.198 0.176 1.26 0.262
## factor(period)5 -0.281 0.202 1.94 0.163
## ---
## 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] 0.112
# 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] 0.278
```