N.pop <- 1000000

# Size of super-population.

prop.treat <- 0.20

# Prevalence of treatment in the super-population.

prop.outcome <- 0.10

# Proportion of subjects with the outcome if everyone was a control.

target.rr <- 0.80

# True population relative risk

tolerance <- 0.0001

# How close is marginal relative risk to the target relative risk

set.seed(12072022)

# Set random number seed for reproducibility.

################################################################################

# Generate 10 baseline covariates for each subject in the super-population.

# The first five are from independent standard normal distibutions.

# The last five are from independent Bernoulli distributions with parameter 0.5.

################################################################################

x1 <- rnorm(N.pop,0,1)

x2 <- rnorm(N.pop,0,1)

x3 <- rnorm(N.pop,0,1)

x4 <- rnorm(N.pop,0,1)

x5 <- rnorm(N.pop,0,1)

x6 <- rbinom(N.pop,1,0.5)

x7 <- rbinom(N.pop,1,0.5)

x8 <- rbinom(N.pop,1,0.5)

x9 <- rbinom(N.pop,1,0.5)

x10 <- rbinom(N.pop,1,0.5)

X <- cbind(1,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10)

# Add a column for an intercept to the matrix of baseline covariates.

################################################################################

# Generate a binary treatment variable with the given prevalence of treatment.

################################################################################

B.treat <- c(log(1.1),log(2),log(3),log(1.5),log(1.5),

log(1.1),log(2),log(3),log(1.5),log(1.5))

# Regression coefficients for treatment-selection model.

treat.function <- function(b0.treat){

# Function for determining prevalence of treatment with a given intercept.

beta.treat.modified <- c(b0.treat,B.treat)

# Set the intercept of the treat model to the given value.

XB <- X %\*% beta.treat.modified

p.treat <- exp(XB)/(1 + exp(XB))

Y <- rbinom(N.pop,1,p.treat)

return(Y)

remove(beta.treat.modified,XB,p.treat,Y)

}

# End of function for estimating prevalence of treatment.

# Use a bisection approach to determine the intercept that results in the

# desired prevalence of treatment.

int.low <- -10

int.high <- 10

iter <- 1

treat.prev <- 1

while(abs(treat.prev - prop.treat) > 0.0001){

set.seed(iter)

int.mid <- (int.low + int.high)/2

treat <- treat.function(b0.treat=int.mid)

treat.prev <- mean(treat)

if (treat.prev < prop.treat) int.low <- int.mid else

int.high <- int.mid

iter <- iter + 1

}

beta0.treat <- int.mid

# Intercept for treatment-selection model.

remove(int.low,int.mid,int.high,iter,treat.prev,treat.function)

################################################################################

# Define regression coefficients for the outcome model.

################################################################################

B.outcome <- c(log(1.25),log(1.5),log(1.75),log(2),log(2.5),

log(1.25),log(1.5),log(1.75),log(2),log(2.5))

# Regression coefficients for the outcome model.

################################################################################

# Generate binary outcomes with the intercept of the outcomes model set

# to a specific value. Generates outcomes under assumption of no treatment.

################################################################################

Y0.function <- function(b0.outcome){

B.outcome.modified <- c(b0.outcome,B.outcome)

# Set the intercept of the outcome model.

XB0 <- X %\*% B.outcome.modified

p.outcome <- exp(XB0)/(1 + exp(XB0))

outcome0 <- rbinom(N.pop,1,p.outcome)

return(cbind(outcome0,XB0))

remove(B.outcome.modified,XB0,p.outcome,outcome0)

}

# End of function for estimating prevalence of the outcome under control.

# Use a bisection approach to determine the intercept that results in the

# desired prevalence of the outcome.

int.low <- -10

int.high <- 10

iter <- 1

outcome.prev <- 1

while(abs(outcome.prev - prop.outcome) > tolerance){

set.seed(iter)

int.mid <- (int.low + int.high)/2

function.results <- Y0.function(b0.outcome=int.mid)

Y0 <- function.results[,1]

LP0 <- function.results[,2]

outcome.prev <- mean(plogis(LP0))

if (outcome.prev < prop.outcome) int.low <- int.mid else

int.high <- int.mid

iter <- iter + 1

}

outcome.intercept <- int.mid

# This is the intercept that results in the desired prevalence of outcome in

# the treated population.

remove(B.outcome,int.low,int.mid,int.high,iter,outcome.prev,Y0.function,

function.results)

##############################################################################

# Determine treatment odds ratio that results in the desired relative risk.

##############################################################################

B.outcome <- c(log(1.25),log(1.5),log(1.75),log(2),log(2.5),

log(1.25),log(1.5),log(1.75),log(2),log(2.5))

# Regression coefficients for the outcome model.

# The magnitude of the effects is in a different order than the treatment model.

B.outcome <- c(outcome.intercept,B.outcome)

# Change the intercept of the outcome model to the value determined above

# that results in the desired prevalence of outcome under control.

################################################################################

# Function for generating binary outcome if everyone were treated.

################################################################################

Y1.function <- function(treat.beta){

# Function for determining relative risk with a given treatment log-odd ratio.

XB1 <- (X %\*% B.outcome) + treat.beta

# The linear predictor under treatment.

p1.outcome <- exp(XB1)/(1 + exp(XB1))

outcome1 <- rbinom(N.pop,1,p1.outcome)

return(cbind(outcome1,XB1))

remove(XB1,p1.outcome,outcome1)

}

################################################################################

# Use a bisection approach to determine the log-odds ratio that results in the

# desired relative risk in the treated subjects.

################################################################################

int.low <- -10

int.high <- 10

iter <- 1

rr.iter <- 1

while(abs(rr.iter - target.rr) > tolerance){

set.seed(iter)

int.mid <- (int.low + int.high)/2

function.results <- Y1.function(treat.beta=int.mid)

Y1 <- function.results[,1]

LP1 <- function.results[,2]

rr.iter <- mean(plogis(LP1)) / mean(plogis(LP0))

if (rr.iter < target.rr) int.low <- int.mid else

int.high <- int.mid

cat(iter,target.rr,int.mid,rr.iter,file="Table2.txt",fill=T,append=T)

iter <- iter + 1

}

remove(B.outcome,int.low,int.mid,int.high,iter,rr.iter,Y1.function,

function.results)

##############################################################################

# Generate an outcome for each subject.

##############################################################################

Y0 <- rbinom(N.pop,1,plogis(LP0))

Y1 <- rbinom(N.pop,1,plogis(LP1))

Y.binary <- treat\*Y1 + (1-treat)\*Y0