N.pop <- 1000000

# Size of super-population.

target.prop.outcome <- 0.1

# Target prevalence of the outcome in the super-population.

tolerance <- 0.0001

# How close is the empirical outcome prevalence required to be to the

# target outcome prevalence.

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.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.

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

# Define a function for generating binary outcomes with a given value of

# the intercept in the outcomes model.

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

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

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

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

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

# Linear predictor.

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

# Probability of the outcome.

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

return(Y)

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

}

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

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

# desired prevalence of the outcome.

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

# Define endpoints of interval to bisect.

int.low <- -10

int.high <- 10

iter <- 1

outcome.prev <- 1

# Initial value of the empirical value of outcome prevalence.

# This can be any value as long as it differs from the target value.

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

set.seed(iter)

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

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

outcome.prev <- mean(outcome)

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

int.high <- int.mid

cat(iter,target.prop.outcome,int.mid,outcome.prev,file="Table1.txt",

fill=T,append=T)

iter <- iter + 1

}

beta0.outcome <- int.mid

# Intercept for the regression model to produce the desired outcome prevalence.

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

B.outcome,X)