Simulation study

Maksim Helmann

```r
## set simulation parameters
n <- 100     # sample size
pz <- 0.2    # probability of Z = 1
alpha0 <- 0  # logit probability of x = 1 in non-smokers (z = 0)
alpha1 <- 1  # log odds ratio of x = 1 in smokers (z = 1) vs non-smokers
beta0 <- -3  # logit prob of y = 1 in non-coffee drinkers (x = 0) and non-smokers (z = 0)
beta1 <- 0
beta2 <- 2
alpha <- 0.05

## generate confounder Z from a binomial distribution
z <- rbinom(n, size = 1, prob = pz)
## compute probability of observing X = 1 from the inverse logit function
px <- exp(alpha0 + alpha1 * z) / (1 + exp(alpha0 + alpha1 * z))
## randomly generate binary variable X from the above probability
x <- rbinom(n, size = 1, prob = px)
## randomly generate binary variable Y from the inverse logistic function
py <- exp(beta0 + beta1 * x + beta2 * z) / (1 + exp(beta0 + beta1 * x + beta2 * z))
y <- rbinom(n, size = 1, prob = py)
dat <- data.frame(lung = y, coffee = x, smoke = z)
```

```r
# very simple example
library(foreach)
# set parameters
n <- 500
m <- 1000
b0 <- 1; b1 <- 2
pz <- 0.2    # probability of Z = 1
alpha0 <- 0  # logit probability of x = 1 in non-smokers (z = 0)
#alpha1 <- 1  # log odds ratio of x = 1 in smokers (z = 1) vs non-smokers
beta0 <- -3  # logit prob of y = 1 in non-coffee drinkers (x = 0) and non-smokers (z = 0)
beta1 <- 0
```

```r
beta2 <- 2

sim.func <- function(alpha1) {

  # repeat simulation 5 times
  indicator.rej <- foreach(i=1:m, .combine=rbind) %do% {
    set.seed(i + 2024)  # set seed for reproducibility
    ## generate confounder Z from a binomial distribution
    z <- rbinom(n, size = 1, prob = pz)
    ## compute probability of observing X = 1 from the inverse logit function
    px <- exp(alpha0 + alpha1 * z) / (1 + exp(alpha0 + alpha1 * z))
    ## randomly generate binary variable X from the above probability
    x <- rbinom(n, size = 1, prob = px)
    ## randomly generate binary variable Y from the inverse logistic function
    py <- exp(beta0 + beta1 * x + beta2 * z) / (1 + exp(beta0 + beta1 * x + beta2 * z))
    y <- rbinom(n, size = 1, prob = py)

    dat <- data.frame(lung = y, coffee = x, smoke = z)

    ## fit unadjusted logistic regression model
    unadj.mod <- glm(lung ~ coffee, data = dat, family = "binomial")
    unadj.coef <- summary(unadj.mod)$coef
    unadj.p <- unadj.coef[2,4]
    unadj.rej <- (unadj.p < alpha) * 1

    ## fit adjusted logistic regression model
    adj.mod <- glm(lung ~ coffee + smoke, data = dat, family = "binomial")
    adj.p <- summary(adj.mod)$coef[2,4]
    adj.rej <- (adj.p < alpha) * 1
    indicator.rej <- c(unadj.rej, adj.rej)
    return(indicator.rej)

  }

  return(colMeans(indicator.rej))

}

a1.0 <- sim.func(0)
a1.1 <- sim.func(1)
a1.2 <- sim.func(2)
```

Table 1: Type I error rate

alpha_1	Adjusted	Unadjusted
0	0.043	0.043
1	0.038	0.218
2	0.041	0.537

```r
library(kableExtra)
alpha_1.c <- c(0, 1, 2)
adj.c <- c(a1.0[2], a1.1[2], a1.2[2])
unadj.c <- c(a1.0[1], a1.1[1], a1.2[1])
type1.df <- data.frame(alpha_1=alpha_1.c, Adjusted=adj.c, Unadjusted=unadj.c)

type1.df %>%
  kbl(caption = "Type I error rate") %>%
  kable_classic_2(full_width = F, position = "center")
```