Simulation_InClass

Aoqi Xie

2024-11-11

Simulation Study

Finish the simulation study with the following scenarios $\alpha_1 = 0$; $\alpha_1 = 1$; $\alpha_1 = 2$ with sample size of 500, simulation size of 1000, and significance level of 0.05.

Compare the Type I error rate between the adjusted model and unadjusted model.

```
# loading packages
library(foreach)
library(knitr)
library(kableExtra)
```

The following is the code for $\alpha_1 = 0$

```
## set simulation parameters
n <- 500     # sample size
pz <- 0.2    # probability of Z = 1
alpha0 <- 0  # logit probability of x = 1 in non-smokers (z = 0)
alpha1 <- 0 # 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
sig_lvl <- 0.05

# repeat simulation 1000 times
simout <- foreach(i=1:1000, .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)

  ## combine three random variables into a data frame
  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
```

```r
  p_value_unadj <- unadj.coef[2, 4]
  reject_unadj <- ifelse(p_value_unadj < 0.05, 1, 0)

  ## fit adjusted logistic regression model
  adj.mod <- glm(lung ~ coffee + smoke, data = dat, family = "binomial")
  adj.coef <- summary(adj.mod)$coef
  p_value_adj <- adj.coef[2, 4]
  reject_adj <- ifelse(p_value_adj < 0.05, 1, 0)

  dec_lst <- c(reject_unadj, reject_adj)
}
alpha10_sim <- simout
```

The following is the code for $\alpha_1 = 1$

```r
## set simulation parameters
n <- 500     # 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
sig_lvl <- 0.05

# repeat simulation 1000 times
simout <- foreach(i=1:1000, .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)

  ## combine three random variables into a data frame
  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
  p_value_unadj <- unadj.coef[2, 4]
  reject_unadj <- ifelse(p_value_unadj < 0.05, 1, 0)

  ## fit adjusted logistic regression model
  adj.mod <- glm(lung ~ coffee + smoke, data = dat, family = "binomial")
  adj.coef <- summary(adj.mod)$coef
  p_value_adj <- adj.coef[2, 4]
  reject_adj <- ifelse(p_value_adj < 0.05, 1, 0)
```

```
  dec_lst <- c(reject_unadj, reject_adj)
}
alpha11_sim <- simout
```

The following is the code for $\alpha_1 = 2$

```
## set simulation parameters
n <- 500     # sample size
pz <- 0.2    # probability of Z = 1
alpha0 <- 0  # logit probability of x = 1 in non-smokers (z = 0)
alpha1 <- 2 # 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
sig_lvl <- 0.05

# repeat simulation 1000 times
simout <- foreach(i=1:1000, .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)

  ## combine three random variables into a data frame
  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
  p_value_unadj <- unadj.coef[2, 4]
  reject_unadj <- ifelse(p_value_unadj < 0.05, 1, 0)

  ## fit adjusted logistic regression model
  adj.mod <- glm(lung ~ coffee + smoke, data = dat, family = "binomial")
  adj.coef <- summary(adj.mod)$coef
  p_value_adj <- adj.coef[2, 4]
  reject_adj <- ifelse(p_value_adj < 0.05, 1, 0)

  dec_lst <- c(reject_unadj, reject_adj)
}
alpha12_sim <- simout
```

```
type1_errors_0 <- colMeans(alpha10_sim)
type1_errors_1 <- colMeans(alpha11_sim)
type1_errors_2 <- colMeans(alpha12_sim)

# Combine the vectors into a matrix or data frame with rows
```

```r
error_rates <- data.frame(
  "Unadjusted Type 1 Error Rate" = c(type1_errors_0[1], type1_errors_1[1], type1_errors_2[1]),
  "Adjusted Type 1 Error Rate" = c(type1_errors_0[2], type1_errors_1[2], type1_errors_2[2])
)

# Optionally, add row names to indicate each scenario
row.names(error_rates) <- c("alpha1 = 0", "alpha1 = 1", "alpha1 = 2")

# Display the table
print(error_rates)
```

```
##            Unadjusted.Type.1.Error.Rate Adjusted.Type.1.Error.Rate
## alpha1 = 0                        0.043                      0.043
## alpha1 = 1                        0.218                      0.038
## alpha1 = 2                        0.537                      0.041
```

```r
error_rates %>%
  kable(caption = "Type 1 Error Rates for Each Scenario") %>%
  kable_styling(full_width = FALSE)
```

Table 1: Type 1 Error Rates for Each Scenario

	Unadjusted.Type.1.Error.Rate	Adjusted.Type.1.Error.Rate
alpha1 = 0	0.043	0.043
alpha1 = 1	0.218	0.038
alpha1 = 2	0.537	0.041