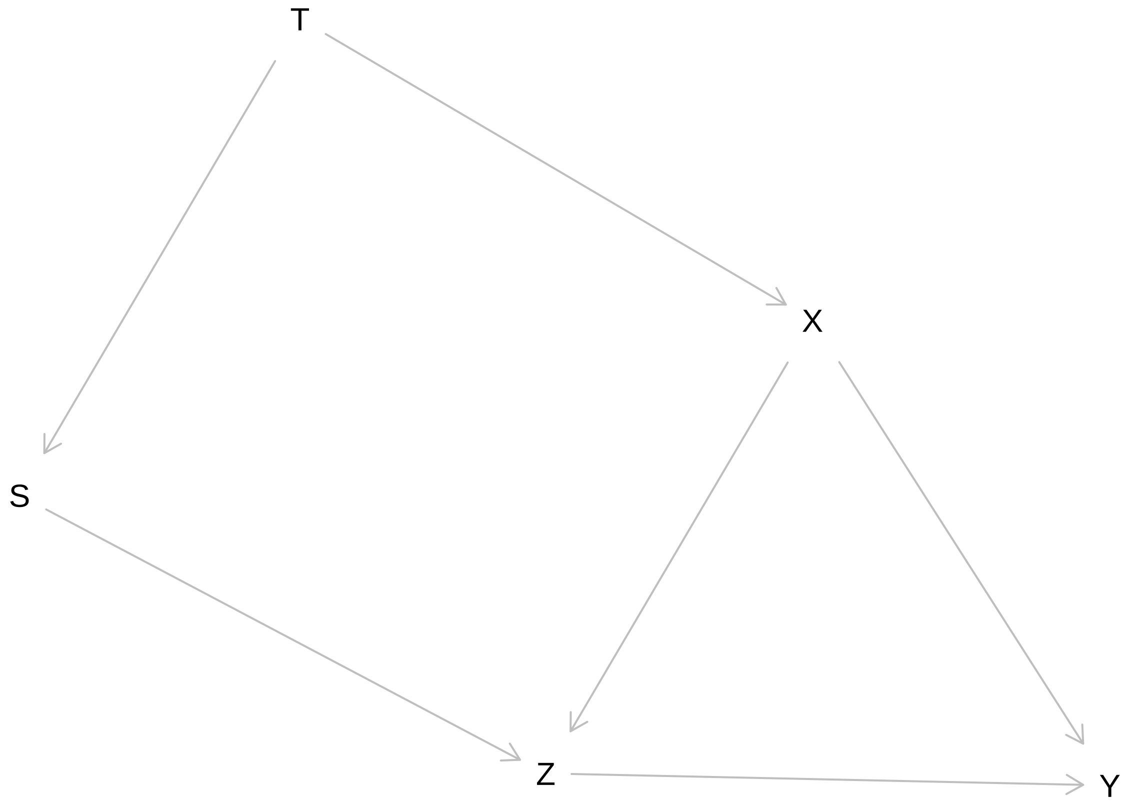
Simulation Study to estimate NDE and NIE

Silpa Soni Nallacheruvu

2025-12-03

Causal dag fig 2(a)



Data generation from the causal dag

```r
generate_data <- function(n) {
  T <- rnorm(n)
  X <- rbinom(n, 1, plogis(0.5 * T))
  S <- 0.5 * T + rnorm(n)
  Z <- rbinom(n, 1, plogis(0.5 * X + 0.5 * S))
  Y <- rbinom(n, 1, plogis(0.5 * X + 0.5 * Z))

  data.frame(Y = Y, X = X, Z = Z, S = S, T = T)
}

sample_data <- generate_data(5)
```

```r
sample_data
```

```
##   Y X Z          S          T
## 1 0 1 1  2.2403748  1.3709584
## 2 0 1 0 -1.4699905 -0.5646982
## 3 0 0 0  0.3333771  0.3631284
## 4 1 1 1 -0.7697013  0.6328626
## 5 1 1 0  1.8155070  0.4042683
```

The true effects of NDE and NIE via Monte Carlo approximation

```r
true_effects <- function(n = 1e6) {

  T <- rnorm(n)
  X_0 <- 0
  X_1 <- 1
  S <- 0.5 * T + rnorm(n)

  Z_0 <- rbinom(n, 1, plogis(0.5 * X_0 + 0.5 * S))
  Z_1 <- rbinom(n, 1, plogis(0.5 * X_1 + 0.5 * S))

  Y_00 <- rbinom(n, 1, plogis(0.5 * X_0 + 0.5 * Z_0))
  Y_01 <- rbinom(n, 1, plogis(0.5 * X_0 + 0.5 * Z_1))
  Y_10 <- rbinom(n, 1, plogis(0.5 * X_1 + 0.5 * Z_0))

  # NDE = E[Y(1, Z(0))] - E[Y(0, Z(0))]
  true_nde <- mean(Y_10) - mean(Y_00)

  # NIE = E[Y(0, Z(1))] - E[Y(0, Z(0))]
  true_nie <- mean(Y_01) - mean(Y_00)

  return(list(true_nde = true_nde, true_nie = true_nie))
}

# Generate large dataset to approximate true effects
true_effects_values <- true_effects()

true_nde <- true_effects_values$true_nde
true_nie <- true_effects_values$true_nie

true_nde
```

```
## [1] 0.115531
```

```r
true_nie
```

```
## [1] 0.014733
```

Estimator NDE

The plug-in estimator for NDE :

$\widehat{\text{NDE}} = \sum_s \sum_z \left[\widehat{E}(Y \mid x, z) - \widehat{E}(Y \mid x^*, z)\right] \widehat{P}(z \mid x^*, s)\, \widehat{P}(s)$

```r
estimate_NDE <- function(data, x = 1, x_star = 0) {
  # Fit models
  model_Y <- glm(Y ~ X + Z, data = data, family = binomial)
  model_Z <- glm(Z ~ X + S, data = data, family = binomial)

  n <- nrow(data)
  nde_sum <- 0

  # By monte carlo approximation, integral over s is approximated by sum over observed s values
  # p(s) is approximated by the empirical distribution of s
  p_s <- 1/n

  for (i in 1:n) {
    s_i <- data$S[i]

    for(z in c(0,1)) {

      # get the predicted p(Z=1|x*,s)
      prob_z1 <- predict(model_Z, newdata = data.frame(X = x_star, S = s_i), type = "response")

      if(z == 1) {
        p_z_xstar <- prob_z1
      } else {
        p_z_xstar <- 1 - prob_z1
      }

      # E[Y|X=x, Z=z] = P(Y=1|x, z)
      y_x <- predict(model_Y, newdata = data.frame(X = x, Z = z), type = "response")

      #  E[Y|X=x_star, Z=z] = P(Y=1|x*, z)
      y_xstar <- predict(model_Y, newdata = data.frame(X = x_star, Z = z), type = "response")

      # Using the plug-in estimator for NDE
      nde_sum <- nde_sum + (y_x - y_xstar)* p_z_xstar * p_s
    }
  }

  return(nde_sum)
}
```

Estimator NIE

The plug-in estimator for NIE :

$\widehat{\mathrm{NIE}} = \sum_s \sum_z \widehat{E}(Y \mid x^*, z) \left[\widehat{P}(z \mid x, s) - \widehat{P}(z \mid x^*, s)\right] \widehat{P}(s)$

```r
estimate_NIE <- function(data, x = 1, x_star = 0) {
  # Fit models
  model_Y <- glm(Y ~ X + Z, data = data, family = binomial)
  model_Z <- glm(Z ~ X + S, data = data, family = binomial)

  n <- nrow(data)
  nie_sum <- 0
```

```
  # By monte carlo approximation, integral over s is approximated by sum over observed s values
  # p(s) is approximated by the empirical distribution of s
  p_s <- 1/n

  for (i in 1:n) {
    s_i <- data$S[i]

    for(z in c(0,1)) {

      # get the predicted p(Z=1|x,s)
      prob_z1 <- predict(model_Z, newdata = data.frame(X = x, S = s_i), type = "response")

      if(z == 1) {
        p_zx <- prob_z1
      } else {
        p_zx <- 1 - prob_z1
      }
      # get the predicted p(Z=1|x*,s)
      prob_z0 <- predict(model_Z, newdata = data.frame(X = x_star, S = s_i), type = "response")

      if(z == 1) {
        p_z_xstar <- prob_z0
      } else {
        p_z_xstar <- 1 - prob_z0
      }

      # E[Y|X=x_star, Z=z] = P(Y=1|x*, z)
      y_z_xstar <- predict(model_Y, newdata = data.frame(X = x_star, Z = z), type = "response")

      # Using the plug-in estimator for NIE
      nie_sum <- nie_sum + y_z_xstar * (p_zx - p_z_xstar) * p_s
    }
  }

  return(nie_sum)
}
```

Run the simulations

```
sample_sizes <- c(100, 500, 1000, 2000, 5000, 10000)
n_sims <- 100

results <- data.frame()

for (n in sample_sizes) {
  cat("Generating data for n =", n, "...\n")

  nde_estimates <- replicate(n_sims, {
    data <- generate_data(n)
    estimate_NDE(data)
  })

  nie_estimates <- replicate(n_sims, {
```

```r
    data <- generate_data(n)
    estimate_NIE(data)
  })

  nde_mean <- mean(nde_estimates)
  nie_mean <- mean(nie_estimates)

  results <- rbind(results, data.frame(
    n = n,
    NDE_mean = nde_mean,
    NDE_bias = nde_mean - true_nde,
    NIE_mean = nie_mean,
    NIE_bias = nie_mean - true_nie
  ))
}
```

```
## Generating data for n = 100 ...
## Generating data for n = 500 ...
## Generating data for n = 1000 ...
## Generating data for n = 2000 ...
## Generating data for n = 5000 ...
## Generating data for n = 10000 ...
```

Display the results

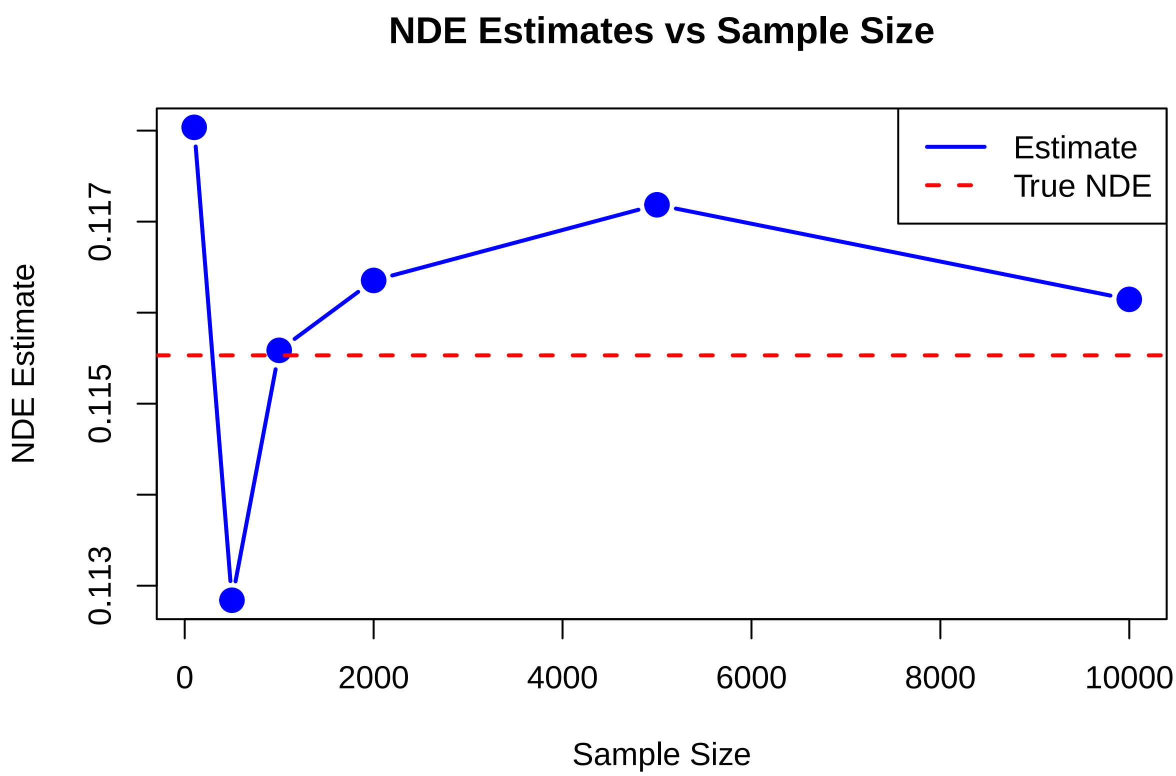
```r
print(results)
```

```
##       n  NDE_mean      NDE_bias   NIE_mean      NIE_bias
## 1   100 0.1180359  2.504945e-03 0.01107250 -0.0036604960
## 2   500 0.1128405 -2.690453e-03 0.01342309 -0.0013099143
## 3  1000 0.1155858  5.483396e-05 0.01507889  0.0003458853
## 4  2000 0.1163541  8.230863e-04 0.01344997 -0.0012830280
## 5  5000 0.1171852  1.654181e-03 0.01378050 -0.0009525036
## 6 10000 0.1161461  6.151436e-04 0.01413779 -0.0005952095
```

Check if the estimators are asymptotically unbiased

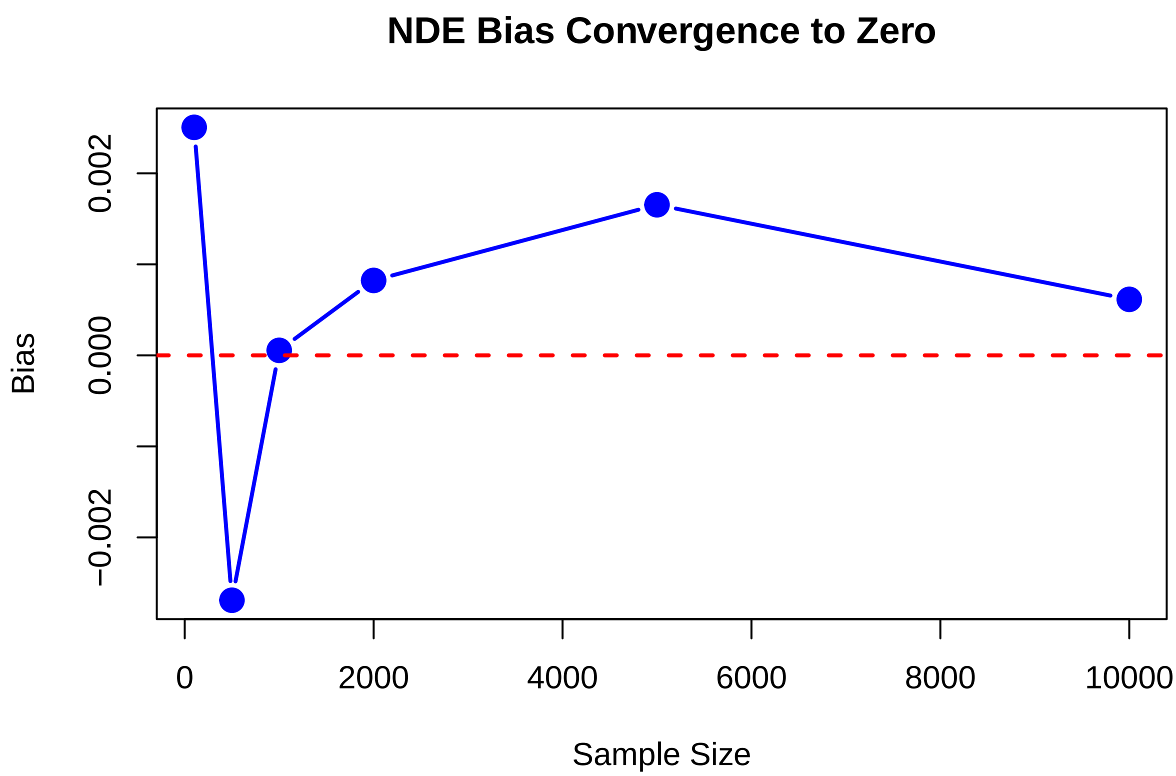
NDE Estimates

```r
plot(results$n, results$NDE_mean,
     type = "b", pch = 19, col = "blue", lwd = 2, cex = 1.5,
     xlab = "Sample Size", ylab = "NDE Estimate",
     main = "NDE Estimates vs Sample Size",
     ylim = range(c(results$NDE_mean, true_nde)))
abline(h = true_nde, col = "red", lty = 2, lwd = 2)
legend("topright", legend = c("Estimate", "True NDE"),
       col = c("blue", "red"), lty = c(1, 2), lwd = 2)
```



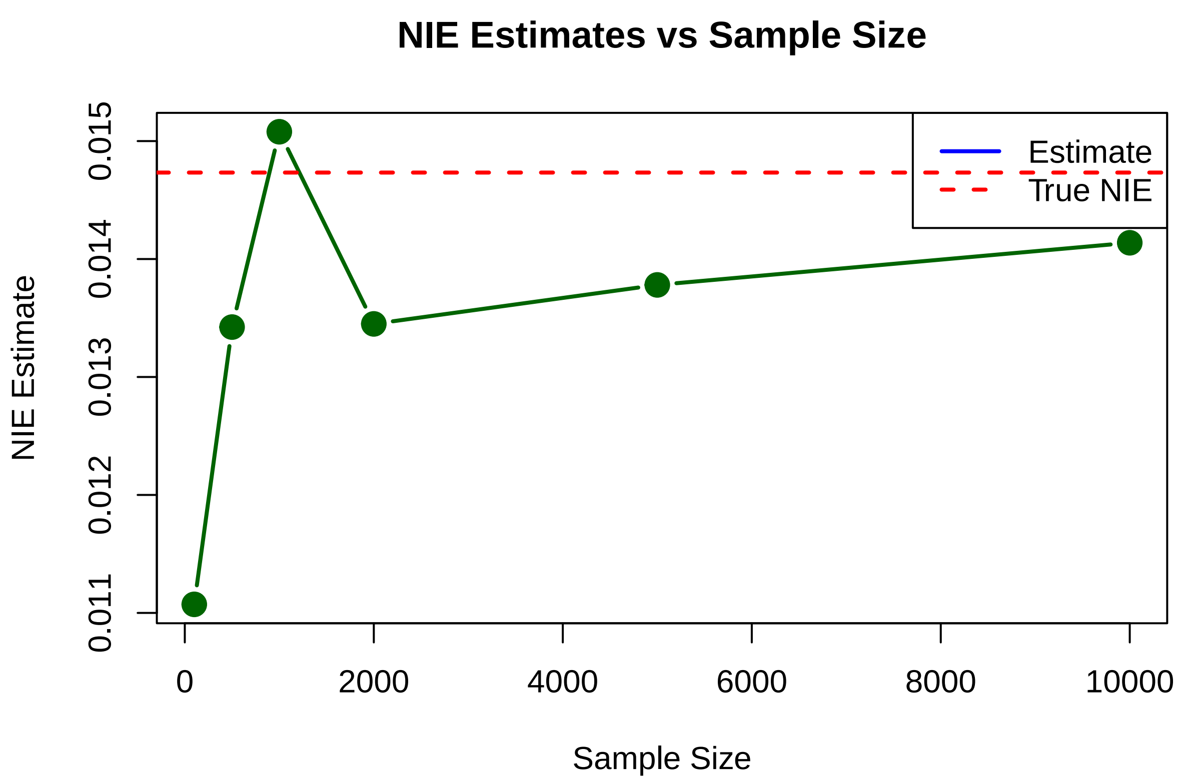
NDE Bias

```
plot(results$n, results$NDE_bias,
     type = "b", pch = 19, col = "blue", lwd = 2, cex = 1.5,
     xlab = "Sample Size", ylab = "Bias",
     ylim = range(c(0, results$NDE_bias)),
     main = "NDE Bias Convergence to Zero")
abline(h = 0, col = "red", lty = 2, lwd = 2)
```



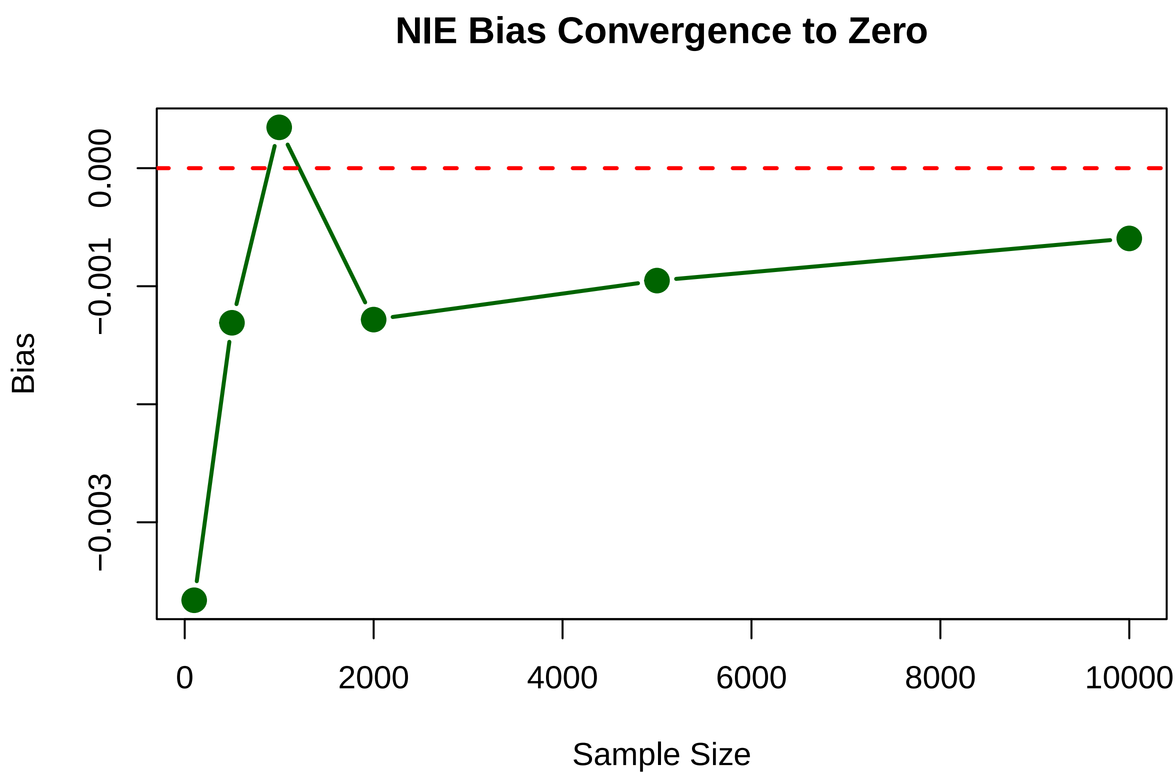
NIE Estimates

```
plot(results$n, results$NIE_mean,
     type = "b", pch = 19, col = "darkgreen", lwd = 2, cex = 1.5,
     xlab = "Sample Size", ylab = "NIE Estimate",
     main = "NIE Estimates vs Sample Size",
     ylim = range(c(results$NIE_mean, true_nie)))
abline(h = true_nie , col = "red", lty = 2, lwd = 2)
legend("topright", legend = c("Estimate", "True NIE"),
       col = c("blue", "red"), lty = c(1, 2), lwd = 2)
```



NIE Bias

```
plot(results$n, results$NIE_bias,
     type = "b", pch = 19, col = "darkgreen", lwd = 2, cex = 1.5,
     xlab = "Sample Size", ylab = "Bias",
     ylim = range(c(0, results$NIE_bias)),
     main = "NIE Bias Convergence to Zero")
abline(h = 0, col = "red", lty = 2, lwd = 2)
```


NIE Bias Convergence to Zero
Bias
0.000
−0.001
−0.003
0
2000
4000
6000
8000
10000
Sample Size