Simulation Study

```
library(ggplot2)
library(tidyverse)
library(ggthemes)
library(latex2exp)
library(patchwork)
library(Matrix)
library(RColorBrewer)
```

Assumed Model

For all simulation studies we assume a randomized trial with no covariates. We assume the outcome is linear model with a term for an individual's own treatment as well as an "interference" term for each other individual it's group. The interference terms are prefixed by a parameter γ that controls the level of interference, where $\gamma = 0$ implies no interference.

$$Y_{ij}(\mathbf{a}) = \beta_0 + \beta_1 a_j + \gamma \sum_{k \neq j} \beta_{kj}^{(i)} a_k + \epsilon_{ij}$$
$$\epsilon_{ij} \sim N(0, \sigma^2 = 0.1)$$

We let the baseline expected level of individual infection when everyone is untreated is $\beta_0 = 10$. We assume an individual experiences a 5 point reduction in their level of infection due to their own treatment with $\beta_1 = -5$. Further, we assume an individual experiences a 0.1 point reduction in level of infection due to the treatment of each other individual with $\beta_{jk}^{(i)} = \beta_2 = -0.1$.

```
beta0 = 10
beta1 = -5
beta2 = -0.1
sigmasq = 0.1
```

Finite Sample Case

In this section, we compute the true population average direct and indirect effects analytically under a finite sample setting (randomization type A). We compare these analytical ground truths to estimates on simulated data.

We implement the finite sample randomization procedure as follows: first, the sampling procedure S_i is chosen for each group i. Then, treatments A_{ij} are assigned according to the assigned sampling procedure.

$$S_i \sim \text{finite sample from } \{\psi, \psi, \psi, \phi, \phi\}$$

$$A_{ij} \sim \begin{cases} \text{finite sample from 30 treated, 70 untreated, without replacement} & \text{if } S_i = \psi \\ \text{finite sample from 50 treated, 50 untreated, without replacement} & \text{if } S_i = \phi \end{cases}$$

```
simulate_data_finite <- function(
  n = 100, # number of subjects per group
  S_values = c(1,1,1,2,2),
  A_options_S1 = c(rep(1, 30), rep(0, 70)),
  A_options_S2 = c(rep(1, 50), rep(0, 50)),
  gamma = 0 # degree of interference
) {
  group_S <- sample(S_values)
  group <- rep(1:length(S_values), each = n)
  S <- rep(group_S, each = n)

  A <- lapply(group_S, function(s) {
    if (s == 1) {
      return(sample(A_options_S1))
    } else {
      return(sample(A_options_S2))
    }
  })

  Y0 <- sapply(1:length(S), function(i) {
    g <- group[i]
    j <- i %% n
    beta0 + gamma * beta2 * sum(A[[g]][-j]) + rnorm(1, 0, sqrt(sigmasq))
  })
  Y1 <- Y0 + beta1
```

```
  A <- do.call(c, A)
  Y <- A * Y1 + (1-A) * Y0

  return(data.frame(
    group = group,
    S = S,
    A = A,
    Y0 = Y0,
    Y1 = Y1,
    Y = Y
  ))
}
```

Ignoring Interference

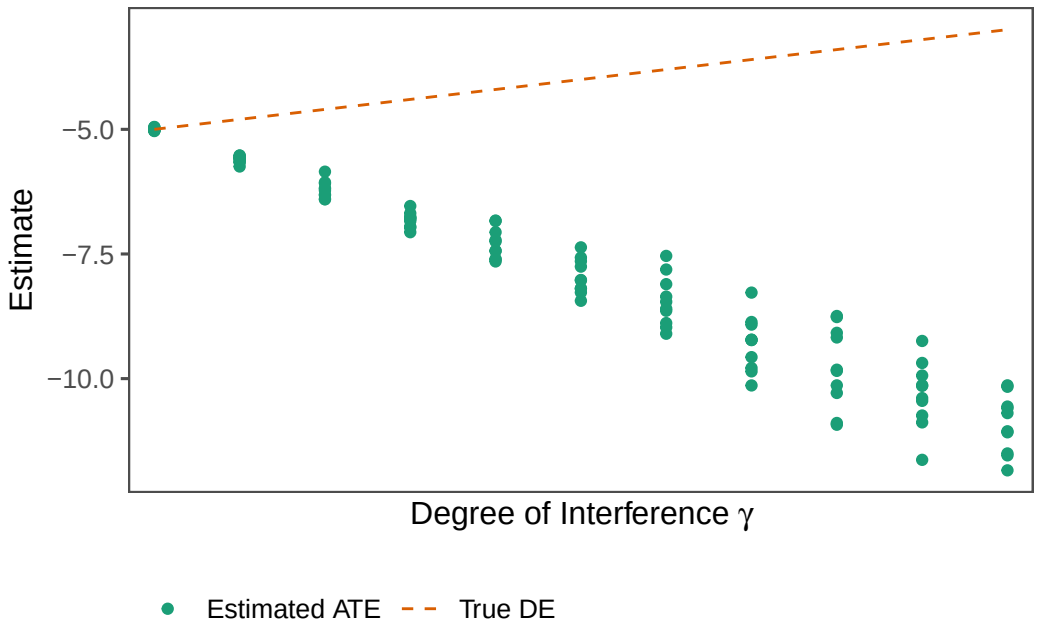
We start with a motivating example of the importance of the interference assumption and how standard methods fail when the assumption is broken but ignored. Suppose we wish to estimate the effect of an individual's vaccination status (treatment) on the severity of disease (outcome). If we ignore any possibility of interference and incorrectly say that the no interference assumption is met, we would compute the ATE and interpret it as a direct effect.

Across varying degrees of interference, we simulated 50 datasets and use the difference in means approach to estimate the ATE. The figure below shows that as the degree of interference γ increases, these naive estimates fall far away from the true direct effect.

One can imagine a more extreme scenario where ignoring interference may mean estimating a null effect or estimating an effect of the wrong direction. In the other direction, as in the example above, the magnitude of the effect may be far overestimated, which would be misleading, for example, in clinical trial results. Clearly, it is necessary to correctly handle interference when it is present by estimating direct and indirect effects separately.

```
Nrep = 10
set.seed(123)
gammas <- seq(0, 20, 2)
results <- matrix(nrow = 0, ncol = 4)
colnames(results) <- c('gamma','PE','IPW','AIPW')

for (k in 1:length(gammas)) {
  gamma = gammas[k]
  for (rep in 1:Nrep) {
```

```
    data <- simulate_data_finite(gamma = gamma)
    outcome_model <- lm(Y~A, data = data)

    phat <- mean(data$A)
    mhat <- outcome_model$fitted.values
    mhat1 <- predict(outcome_model, newdata = data %>% mutate(A = 1))
    mhat0 <- predict(outcome_model, newdata = data %>% mutate(A = 0))

    PE <- mean(mhat1) - mean(mhat0)
    IPW <- mean(data$Y*data$A / phat) - mean(data$Y*(1-data$A) / (1-phat))
    AIPW <- mean(
      (data$A / phat - (1-data$A) / (1-phat)) * (data$Y - mhat) +
      mhat1 - mhat0
    )
    results <- rbind(results, c(gamma, PE, IPW, AIPW))
  }
}

true = data.frame(gamma = gammas, y = beta1 - beta2*gammas, Estimator = "True DE")

data.frame(results) %>%
  pivot_longer(cols = c('PE','IPW','AIPW'), names_to = 'Estimator', values_to = 'Estimate'
  filter(Estimator == 'PE') %>%
  mutate(Estimator = 'Estimated ATE') %>%
  ggplot(aes(x = gamma, y = Estimate, color = Estimator)) +
  geom_point() +
  labs(x = TeX('Degree of Interference $\\gamma$'),
       y = 'Estimate') +
  theme_few() +
  scale_color_brewer(palette = "Dark2") +
  theme(plot.title = element_blank(), legend.position = 'bottom', legend.justification = '
  geom_line(data = true, aes(x = gamma, y = y), linetype = 'dashed') +
  scale_x_discrete(breaks = gammas, labels = as.character(gammas))
```




```
ggsave("naive_estimates_FINITE.png", height =5, width = 8)
```

Estimating Direct and Indirect Effects

Hudgens and Halloran (2008) defines the standard formulation of the population average direct effect as $\bar{DE}(\phi) = \bar{Y}(1|\phi) - \bar{Y}(0|\phi) = \frac{1}{N}\sum_{i=1}^{N} \bar{DE}_i(\phi)$. Below, we compute analytically what this true direct effect is. Define K_ϕ as the fixed number of individuals treated under assignment strategy ϕ versus K_ψ as the number under assignment strategy ψ.

$$
\begin{aligned}
\bar{Y}_{ij}(a|\phi) &= \sum_{\mathbf{s}\in\mathcal{A}(n_i-1)} Y_{ij}(\mathbf{a}_{i(j)}=\mathbf{s}, a_{ij}=a)\cdot Pr_{\phi}(\mathbf{A}_{i(j)}=\mathbf{s}|A_{ij}=a) \\
&= \sum_{\mathbf{s}\in\mathcal{A}(n_i-1)} \left(\beta_0+\beta_1 a+\gamma\beta_2\sum_{k\neq j}s_k+\epsilon_{ij}\right)\cdot\frac{1}{\binom{n-1}{K_\phi-a}}\cdot I(\sum_{k\neq j}s_k=K_\phi-a) \\
&= \left(\beta_0+\beta_1 a_j+\gamma\beta_2(K_\phi-a)+\epsilon_{ij}\right)\cdot\frac{\binom{n-1}{K_\phi-a}}{\binom{n-1}{K_\phi-a}} \\
&= \beta_0+\beta_1 a+\gamma\beta_2(K_\phi-a)+\epsilon_{ij} \\
\bar{Y}_i(a|\phi) &= \frac{1}{n_i}\sum_{j=1}^{n_i} Y_{ij}(a|\phi) = \beta_0+\beta_1 a+\gamma\beta_2(K_\phi-a)+\frac{1}{n_i}\sum_{j=1}^{n_i}\epsilon_{ij} \\
\bar{DE}_i(\phi) &= \bar{Y}_i(1|\phi)-\bar{Y}_i(0|\phi) = \beta_1-\gamma\beta_2 \\
\bar{DE}(\phi) &= \frac{1}{N}\sum_{i=1}^{N}\bar{DE}_i(\phi) = \beta_1-\gamma\beta_2
\end{aligned}
$$

We see that the true population average direct effect does not depend on K_ϕ, so will be identical for assignment strategy ψ. Further, we perhaps unexpectedly see that this "direct effect" depends on the degree of interference γ and the interference coefficient β_2. This is the problem identified in VanderWeele and Tchetgen Tchetgen (2011) that will be remedied in the following section using Randomization Type B.

Hudgens and Halloran (2008) defines the standard formulation of the population average indirect effect as $\bar{IE}(\phi,\psi)=\bar{Y}(0|\phi)-\bar{Y}(0|\psi)$. We compute analytically what this true direct effect is under a finite sample setting and our defined data generating function.

$$
\begin{aligned}
\bar{Y}(a|\phi) &= \frac{1}{N}\sum_{i=1}^{N}\bar{Y}_i(a|\phi) = \beta_0+\beta_1 a+\gamma\beta_2(K_\phi-a)+\frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n-i}\epsilon_{ij} \\
\bar{IE}(\phi,\psi) &= \bar{Y}(0|\phi)-\bar{Y}(0|\psi) = \gamma\beta_2(K_\phi-K_\psi)
\end{aligned}
$$

In our simulation studies, we use the standard estimators from Hudgens and Halloran (2008) based on $\hat{Y}_i(a|\phi)=\frac{\sum_{j=1}^{n_i} I(A_{ij}=a)Y_{ij}(\mathbf{A}_i)}{\sum_{j=1}^{n_i} I(A_{ij}=a)}$. The first figure below shows estimated population average direct effects across 100 simulated datasets for each value of degree of interference γ. Estimates are centered at our empirically derived true population average direct effect of $\beta_1-\gamma\beta_2=-5+0.1\gamma$. We also see that the variability of these estimates increase with the degree of interference.

We see that the true population average indirect effect only depends on the difference between the assignment strategies, as one would expect. The second figure below shows estimated values across 100 datasets for each value of degree of interference γ. Estimates are centered at

the true population average indirect effect $\gamma\beta_2(K_\phi - K_\psi) = -2\gamma$. This negative value makes sense: we expect a protective effect of vaccinating 50 individuals versus only 30.

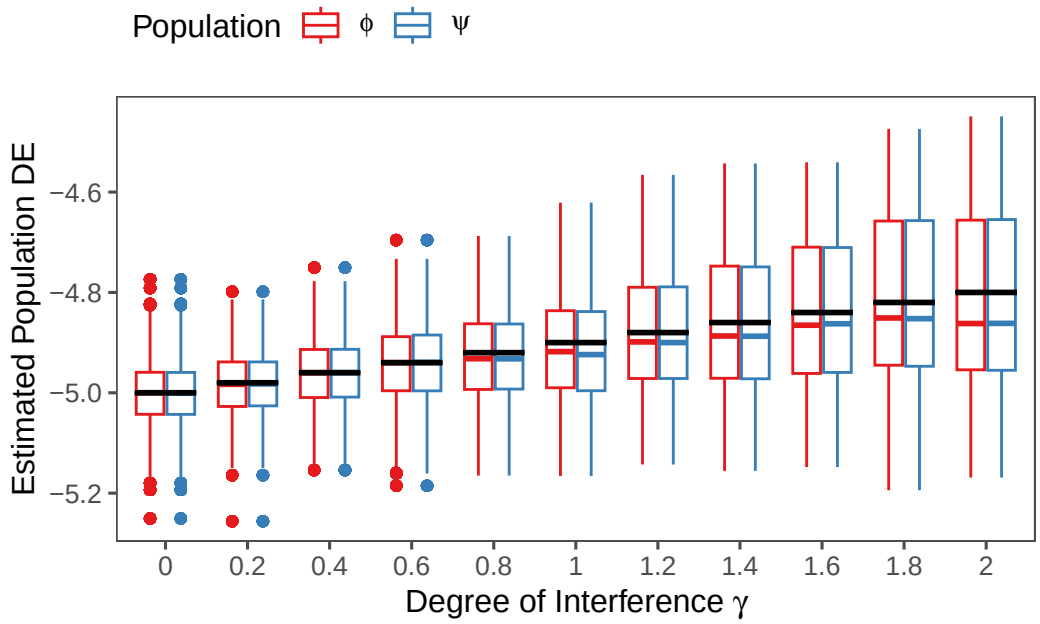
```
Nrep = 100
set.seed(123)
gammas <- seq(0, 2, 0.2)

example_data <- simulate_data_finite(gamma = gamma)
col_names <- c(
  "gamma",
  paste0("group", unique(example_data$group), "S"),
  paste0("group", unique(example_data$group), "DE"),
  paste0("pop", unique(example_data$S), "DE"),
  "popIE", "popTE", "popOE"
)

results <- matrix(nrow = 0, ncol = length(col_names))
colnames(results) <- col_names

for (k in 1:length(gammas)) {
  gamma = gammas[k]
  for (rep in 1:Nrep) {
    data <- simulate_data_finite(gamma = gamma)

    # group results: Y1bar, Y0bar, group direct effects
    group_results <- matrix(nrow = 0, ncol = 5)
    colnames(group_results) <- c('group','Ybar1','Ybar0','DE','S')
    for (g in unique(data$group)) {
      group_Y1 <- mean(data[data$group == g & data$A == 1, 'Y'])
      group_Y0 <- mean(data[data$group == g & data$A == 0, 'Y'])
      group_DE <- group_Y1 - group_Y0
      group_S <- unique(data[data$group == g, 'S'])
      group_results <- rbind(group_results, c(g, group_Y1, group_Y0, group_DE, group_S))
    }
    group_results <- data.frame(group_results)

    # population results: Y1bar, Y0bar, population direct effects
    population_results <- group_results %>%
      group_by(S) %>%
      summarize(
        S = first(S),
        Ybar0 = mean(Ybar0),
```

```
        Ybar1 = mean(Ybar1)
      ) %>%
      mutate(
        DE = Ybar1 - Ybar0
      )

    # indirect effect
    pop_IE <- population_results %>% filter(S == 1) %>% pull(Ybar0) -
              population_results %>% filter(S == 2) %>% pull(Ybar0)

    # direct effect
    pop_TE <- population_results %>% filter(S == 1) %>% pull(Ybar1) -
              population_results %>% filter(S == 2) %>% pull(Ybar0)

    # overal results
    overall_results <- data %>%
      group_by(S, group) %>%
      summarize(
        Ybar = mean(Y),
        .groups = 'keep'
      ) %>%
      group_by(S) %>%
      summarize(
        Ybar = mean(Ybar)
      )

    # overall effect
    pop_OE <- overall_results %>% filter(S == 1) %>% pull(Ybar) -
              overall_results %>% filter(S == 2) %>% pull(Ybar)

    # check decomposition works!
    stopifnot(abs(pop_IE + population_results$DE[1] - pop_TE) < 0.01)

    results <- rbind(
      results,
      c(gamma, group_results$S, group_results$DE, population_results$DE, pop_IE, pop_TE, p
    )
  }
}
results = data.frame(results)
```

```
groups <- results %>%
  select(1:6) %>%
  pivot_longer(2:6, names_to = 'group', values_to = 'S') %>%
  mutate(
    group = as.numeric(str_extract(group, '\\d+'))
  ) %>%
  merge(
    results %>%
      select(1, 7:11) %>%
      pivot_longer(2:6, names_to = 'group', values_to = 'DE') %>%
      mutate(
        group = as.numeric(str_extract(group, '\\d+'))
      )
  ) %>%
  mutate(
    true_DE = beta1 - beta2*gamma
  )

dark2_palette <- brewer.pal(n = 2, name = "Set1")
```

```
Warning in brewer.pal(n = 2, name = "Set1"): minimal value for n is 3, returning requested pa
```

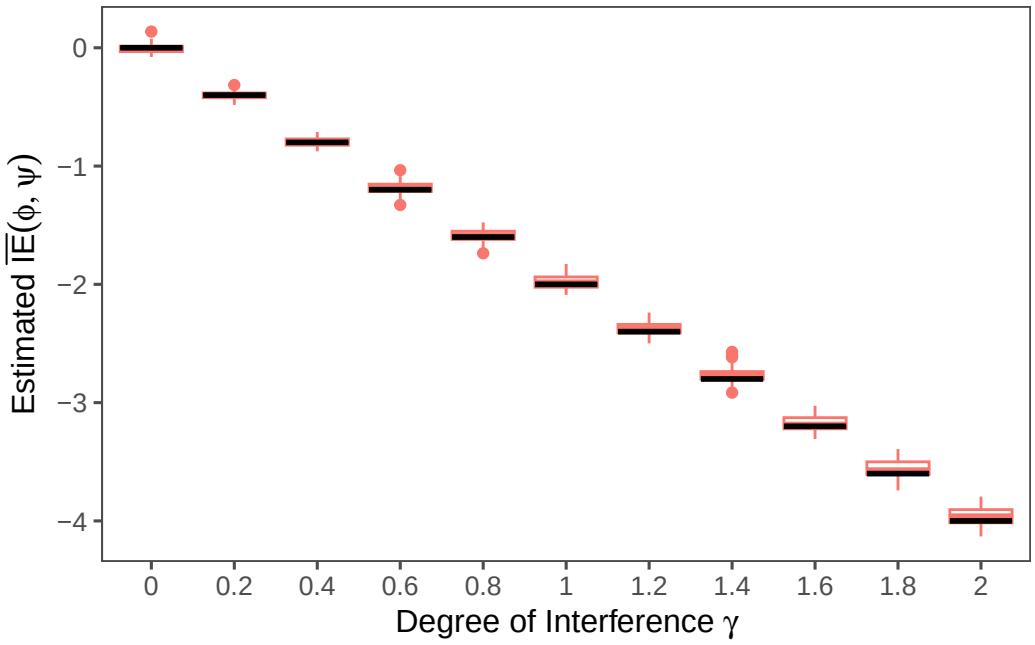
```
groups %>%
  ggplot(aes(x = as.factor(gamma), y = DE, color = as.factor(S))) +
  geom_boxplot() +
  geom_boxplot(aes(x = as.factor(gamma), y = true_DE), color = 'black') +
  labs(x = TeX('Degree of Interference $\\gamma$'),
       y = 'Estimated Population DE',
       color = "Population") +
  theme_few() +
  theme(plot.title = element_blank(), legend.position = 'top', legend.justification = 'lef
  scale_color_manual(values = dark2_palette,
                     labels = c("1" = TeX('$\\phi$'), "2" = TeX('$\\psi$')))
```




```
ggsave("pop_estimates_FINITE.png", height = 5, width = 8)
```

```
populations <- results %>%
  select(c("gamma","pop1DE","pop2DE","popIE","popTE","popOE")) %>%
  pivot_longer(
    cols = c("pop1DE","pop2DE", "popIE", "popTE", "popOE"),
    names_to = "Estimator",
    values_to = "Estimate"
  )

true = data.frame(gamma = gammas, y = -20*beta2*gammas, Estimator = TRUE)

populations %>%
  filter(Estimator == 'popIE') %>%
  ggplot(aes(x = as.factor(gamma), y = -Estimate, color = Estimator)) +
  geom_boxplot() +
  geom_boxplot(data = true, aes(x = as.factor(gamma), y = -y), color = 'black') +
  theme_few() +
  labs(
    x = TeX('Degree of Interference $\\gamma$'),
    y = TeX('Estimated $\\bar{IE}(\\phi, \\psi)$'),
  ) +
```

```
  theme(legend.position = 'none')
```




```
ggsave("IE_estimates_FINITE.png", height = 5, width = 8)
```

Bernoulli Case

In this section, we consider a Bernoulli randomization setting (randomization type B). Importantly, the redefined direct effect version from VanderWeele and Tchetgen Tchetgen (2011) which results in a more intuitive "direct effect" that does not depend on interference at all. We analytically derive the true value of this new direct effect and compare to estimates on simulated data.

We implement the Bernoulli randomization procedure identically to the fixed sample randomization procedure, but this time A_{ij}s are sampled independently rather than from a finite population.

$$S_i \sim \text{finite sample from } \{\psi, \psi, \psi, \phi, \phi\}$$
$$A_{ij} \sim \begin{cases} \text{Bernoulli}(0.3) & \text{if } S_i = \psi \\ \text{Bernoulli}(0.5) & \text{if } S_i = \phi \end{cases}$$

```r
true_ybar_aij <- function(a, gamma, ni, p = 0.3) {
  sum(sapply(0:(ni-1), function(s) {
    choose(ni-1, s) * (beta0 + beta1*a + gamma*s*beta2) * p^s * (1-p)^(ni-1-s)
  }))
}
simulate_data_bernoulli <- function(
    n = 100, # number of subjects per group
    S_values = c(1,1,1,2,2),
    S_probs = c(0.3, 0.5),
    gamma = 0,
    sigmasq = 0.1 # variance of error term
) {
  group_S <- sample(S_values)
  group <- rep(1:length(S_values), each = n)
  S <- rep(group_S, each = n)
  A <- lapply(group_S, function(s) {
    sample(c(0, 1), size = n, replace = TRUE,
           prob = c(1 - S_probs[s], S_probs[s]))
  })

  Y0 <- sapply(1:length(S), function(i) {
    true_ybar_aij(0, gamma, n, S_probs[S[i]]) + rnorm(1, 0, sqrt(sigmasq))
  })
  Y1 <- sapply(1:length(S), function(i) {
    true_ybar_aij(1, gamma, n, S_probs[S[i]])
  })

  A <- do.call(c, A)
  Y <- A * Y1 + (1-A) * Y0

  return(data.frame(
    group = group,
    S = S,
    A = A,
    Y0 = Y0,
    Y1 = Y1,
    Y = Y
  ))
}
```

Estimating Direct and Indirect Effects

Define p_ϕ as the probability of being assigned treatment under assignment strategy ϕ.

$$\begin{aligned}
\bar{Y}^*_{ij}(a|\phi, a') &= \sum_{\in \psi^{n_i-1}} Y_{ij}(\mathbf{a}_{i(j)} = \omega, a_{ij} = a) Pr_\psi(\mathbf{A}_i = \ |a_{ij} = a') \\
&= \sum_{\in \psi^{n_i-1}} \left(\beta_0 + \beta_1 a + \gamma\beta_2 \sum_{k \neq j} \omega_k + \epsilon_{ij} \right) \cdot p_\phi^{\sum \omega} (1 - p_\phi)^{n_i - \sum \omega} \cdot I(\omega_j = a') \\
\bar{DE}^*_{ij}(\psi, a) &= \bar{Y}^*_{ij}(1|\psi, a) - \bar{Y}^*_{ij}(0|\psi, a) = \beta_1 \\
\bar{DE}^*(\psi) &= \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \bar{DE}^*_{ij}(\psi, a) = \beta_1
\end{aligned}$$

We see that the alternate definition of the true population average direct effect does not depend on the degree of interference γ or the interference coefficient β_2. This makes more sense as a "direct effect" because it remains constant regardless of interference.

$$\begin{aligned}
\bar{Y}^*_{ij}(a|\phi) &= \sum_{\in \phi^{n_i}} Y_{ij}(\mathbf{a}_{i(j)} = \ _{(j)}, a_{ij} = a) Pr_\phi(\mathbf{A}_i = \ ) \\
&= \sum_{\in \phi^{n_i}} \left(\beta_0 + \beta_1 a + \gamma\beta_2 \sum_{k \neq j} \omega_k + \epsilon_{ij} \right) \cdot p_\phi^{\sum \omega} (1 - p_\phi)^{n_i - \sum \omega} \\
&= \sum_{t=0}^{n} \left(\beta_0 + \beta_1 a + \gamma\beta_2 t + \epsilon_{ij} \right) \cdot p_\phi^t (1 - p_\phi)^{n_i - t} \text{ where } t = \sum \omega \\
\bar{IE}^*_{ij}(\phi, \psi) &= \bar{Y}^*_{ij}(0|\phi) - \bar{Y}^*_{ij}(0|\psi) \\
&= \sum_{t=0}^{n} \left(\beta_0 + \beta_1 + \gamma\beta_2 t + \epsilon_{ij} \right) \cdot p_\phi^t (1 - p_\phi)^{n_i - t} - \sum_{t=0}^{n} \left(\beta_0 + \gamma\beta_2 t + \epsilon_{ij} \right) \cdot p_\phi^t (1 - p_\phi)^{n_i - t} \\
&= \gamma\beta_2 \cdot (E_{t_\phi}[t] - E_{t_\psi}[t]) \text{ where } t_\phi \sim \text{Binomial}(n_i, p_\phi) \text{ and } t_\psi \sim \text{Binomial}(n_i, p_\psi) \\
&= \gamma n_i \beta_2 (p_\phi - p_\psi) \\
\bar{IE}^*(\phi, \psi) &= \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \bar{IE}^*_{ij}(\phi, \psi) = \gamma n \beta_2 (p_\phi - p_\psi) \text{ assuming } n_i = n \forall i
\end{aligned}$$

The indirect effect is the same as before because $K_\phi = np_\phi$ and $K_\psi = np_\psi$.

In our simulation studies, we use the estimators for this alternative form of direct and indirect effects from VanderWeele and Tchetgen Tchetgen (2011) that are based on $\hat{Y}_i(a|\phi) = \frac{\sum_{j=1}^{n_i} I(A_{ij}=a) Y_{ij}(\mathbf{A}_i)}{\sum_{j=1}^{n_i} I(A_{ij}=a)}$, which align with the finite sample estimators from Hudgens and Halloran

(2008). The first figure below shows that the simulation studies estimate this value very well. Further we see that the variability of these estimates is stable as degree of interference increases. The second figure mirrors that from the previous section because the indirect effect is the same under the original and new formulations.

```
Nrep = 100
set.seed(123)
gammas <- seq(0, 2, 0.2)

example_data <- simulate_data_bernoulli(gamma = gamma)
col_names <- c(
  "gamma",
  paste0("group", unique(example_data$group), "S"),
  paste0("group", unique(example_data$group), "DE"),
  paste0("pop", unique(example_data$S), "DE"),
  "popIE", "popTE", "popOE"
)

results <- matrix(nrow = 0, ncol = length(col_names))
colnames(results) <- col_names

for (k in 1:length(gammas)) {
  gamma = gammas[k]
  for (rep in 1:Nrep) {
    data <- simulate_data_bernoulli(gamma = gamma)

    # group results: Y1bar, Y0bar, group direct effects
    group_results <- matrix(nrow = 0, ncol = 5)
    colnames(group_results) <- c('group','Ybar1','Ybar0','DE','S')
    for (g in unique(data$group)) {
      group_Y1 <- mean(data[data$group == g & data$A == 1, 'Y'])
      group_Y0 <- mean(data[data$group == g & data$A == 0, 'Y'])
      group_DE <- group_Y1 - group_Y0
      group_S <- unique(data[data$group == g, 'S'])
      group_results <- rbind(group_results, c(g, group_Y1, group_Y0, group_DE, group_S))
    }
    group_results <- data.frame(group_results)

    # population results: Y1bar, Y0bar, population direct effects
    population_results <- group_results %>%
      group_by(S) %>%
      summarize(
```

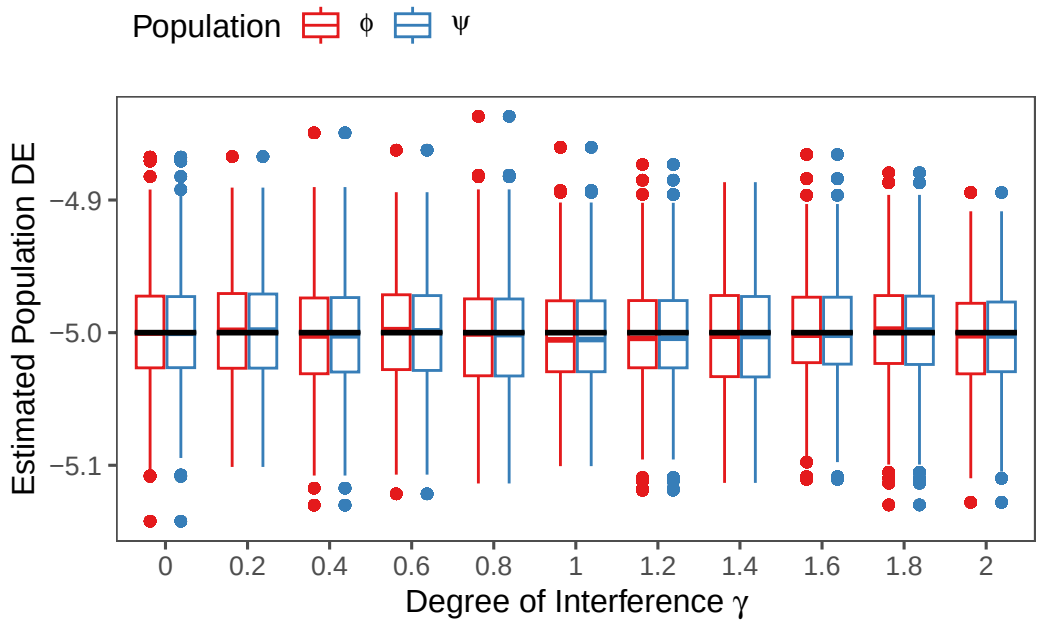
```
      S = first(S),
      Ybar0 = mean(Ybar0),
      Ybar1 = mean(Ybar1)
    ) %>%
    mutate(
      DE = Ybar1 - Ybar0
    )

  # indirect effect
  pop_IE <- population_results %>% filter(S == 1) %>% pull(Ybar0) -
            population_results %>% filter(S == 2) %>% pull(Ybar0)

  # direct effect
  pop_TE <- population_results %>% filter(S == 1) %>% pull(Ybar1) -
            population_results %>% filter(S == 2) %>% pull(Ybar0)

  # overal results
  overall_results <- data %>%
    group_by(S, group) %>%
    summarize(
      Ybar = mean(Y),
      .groups = 'keep'
    ) %>%
    group_by(S) %>%
    summarize(
      Ybar = mean(Ybar)
    )

  # overall effect
  pop_OE <- overall_results %>% filter(S == 1) %>% pull(Ybar) -
            overall_results %>% filter(S == 2) %>% pull(Ybar)

  # check decomposition works!
  stopifnot(abs(pop_IE + population_results$DE[1] - pop_TE) < 0.01)

  results <- rbind(
    results,
    c(gamma, group_results$S, group_results$DE, population_results$DE, pop_IE, pop_TE, p
  )
 }
}
```

```
results = data.frame(results)
```

```
groups <- results %>%
  select(1:6) %>%
  pivot_longer(2:6, names_to = 'group', values_to = 'S') %>%
  mutate(
    group = as.numeric(str_extract(group, '\\d+'))
  ) %>%
  merge(
    results %>%
      select(1, 7:11) %>%
      pivot_longer(2:6, names_to = 'group', values_to = 'DE') %>%
      mutate(
        group = as.numeric(str_extract(group, '\\d+'))
      )
  ) %>%
  mutate(
    true_DE = beta1
  )

dark2_palette <- brewer.pal(n = 2, name = "Set1")
```

```
Warning in brewer.pal(n = 2, name = "Set1"): minimal value for n is 3, returning requested pa
```

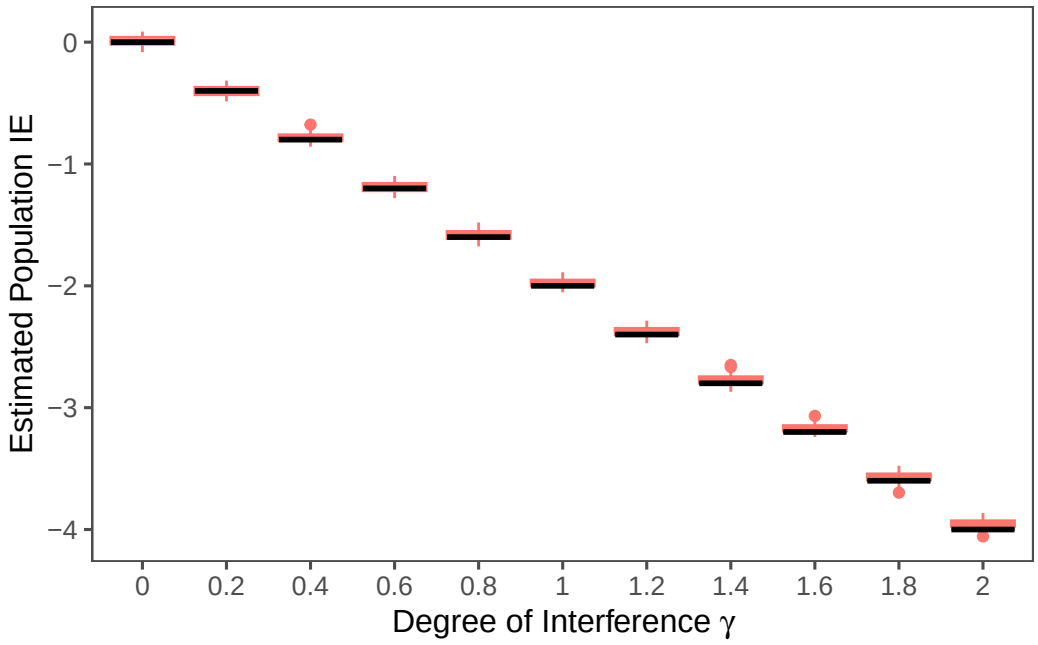
```
groups %>%
  ggplot(aes(x = as.factor(gamma), y = DE, color = as.factor(S))) +
  geom_boxplot() +
  geom_boxplot(aes(x = as.factor(gamma), y = true_DE), color = 'black') +
  labs(x = TeX('Degree of Interference $\\gamma$'),
       y = 'Estimated Population DE',
       color = "Population") +
  theme_few() +
  theme(plot.title = element_blank(), legend.position = 'top', legend.justification = 'lef
  scale_color_manual(values = dark2_palette,
                     labels = c("1" = TeX('$\\phi$'), "2" = TeX('$\\psi$')))
```




```
ggsave("pop_estimates_BERNOULLI.png", height = 5, width = 8)
```

```
populations <- results %>%
  select(c("gamma","pop1DE","pop2DE","popIE","popTE","popOE")) %>%
  pivot_longer(
    cols = c("pop1DE","pop2DE", "popIE", "popTE", "popOE"),
    names_to = "Estimator",
    values_to = "Estimate"
  )

true = data.frame(gamma = gammas, y = -20*beta2*gammas, Estimator = TRUE)

populations %>%
  filter(Estimator == 'popIE') %>%
  ggplot(aes(x = as.factor(gamma), y = -Estimate, color = Estimator)) +
  geom_boxplot() +
  geom_boxplot(data = true, aes(x = as.factor(gamma), y = -y), color = 'black') +
  theme_few() +
  labs(
    x = TeX('Degree of Interference $\\gamma$'),
    y = 'Estimated Population IE',
  ) +
```

```
  theme(legend.position = 'none')
```




```
ggsave("IE_estimates_BERNOULLI.png", height = 5, width = 8)
```


Hudgens, Michael G, and M. Elizabeth Halloran. 2008. "Toward Causal Inference With Interference." *Journal of the American Statistical Association* 103 (482): 832–42. https://doi.org/10.1198/016214508000000292.

VanderWeele, Tyler J., and Eric J. Tchetgen Tchetgen. 2011. "Effect Partitioning Under Interference in Two-Stage Randomized Vaccine Trials." *Statistics & Probability Letters* 81 (7): 861–69. https://doi.org/10.1016/j.spl.2011.02.019.