Break the Simulation 1

Grant Jackson

Break the Simulation #1

Describe your thinking

In order to break the simulation, I ... assigned treatment based on whether x is greater than 5, thereby violating the randomness required to find true causal effect. I also changed the error term to be generated on the treatment status, which introduces bias as the treated units have different distributions than the untreated units.

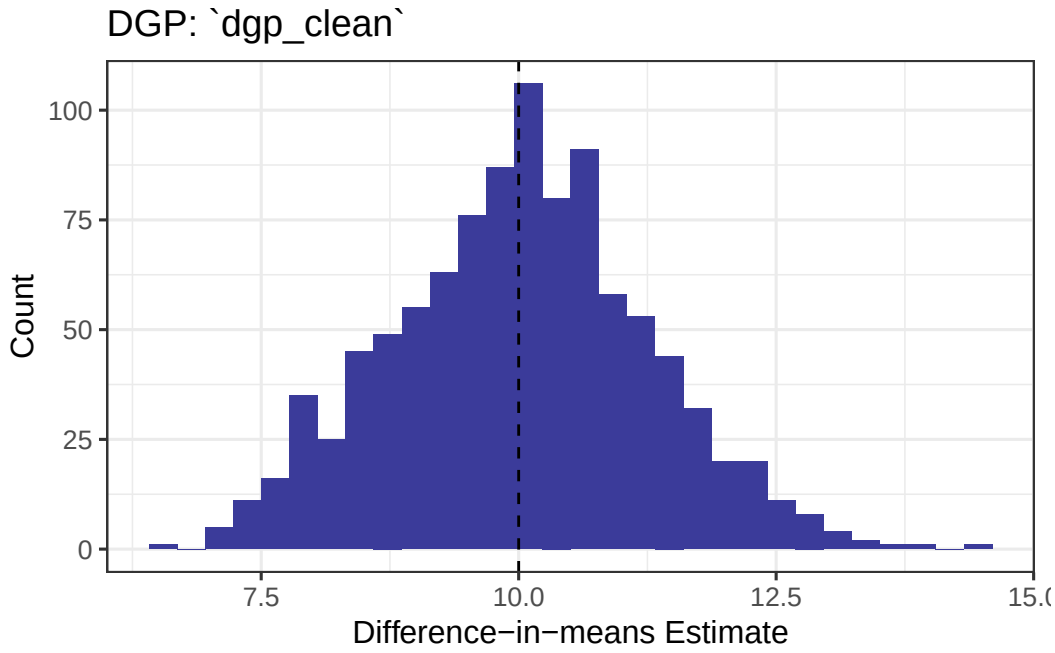
Original Simulation

```
# This is the original code and should not be changed
dgp_clean <- function(true_te = 10) {
  n <- 100
  x <- runif(n = n, 0, 10)

  d <- as.numeric(runif(n = n, 0, 1) > 0.5)

  u <- rnorm(n = n, mean = 0, sd = 2)

  y <- d * true_te + x * 2 + u

  df <- data.frame(
    x = x, treat = d, y = y
  )
}
```

```r
check_df <- function(df) {
  # Assertions to make sure your modifications are not going to break this
  df_has_correct_columns <- all(c("treat", "y") %in% colnames(df))
  assert_that(df_has_correct_columns, msg = "`df` must contain the columns `treat` and `y`.")

  is_treatment_dummy <- all(c(0, 1) %in% unique(df$treat))
  assert_that(is_treatment_dummy, msg = "`df$d` must be a 0/1 treatment indicator variable.")
}
estimate_diff_in_means <- function(df) {
  check_df(df)

  # difference-in-means estimator
  est <- feols(
    y ~ 1 + i(treat, ref = 0), data = df
  )
  coef(est)["treat::1"]
}
```

```r
true_te <- 10

# This is what the monte carlo simulation looks when the dgp satisfies the assumptions of the
set.seed(20240917)
mc_dgp_clean <- purrr::map_dbl(1:1000, function(i) {
  df <- dgp_clean(true_te = true_te)
  estimate_diff_in_means(df)
})

ggplot() +
  geom_histogram(aes(x = mc_dgp_clean), fill = "#3b3b9a") +
  geom_vline(xintercept = true_te, linetype = "dashed") +
  labs(
    title = "DGP: `dgp_clean`",
    x = "Difference-in-means Estimate", y = "Count"
  ) +
  theme_bw(base_size = 12)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Breaking the Simulation

This is the *only* code block that you should modify!

```
dgp_broken <- function(true_te = 10) {
  n <- 100
  x <- runif(n = n, 0, 10)

  d <- as.numeric(x > 5)

  u <- rnorm(n = n, mean = d, sd = 2)

  y <- d * true_te + x * 2 + u

  df <- data.frame(
    x = x, treat = d, y = y
  )
}

df <- dgp_broken()
```

```r
true_te <- 10

set.seed(20240917)
mc_dgp_broken <- purrr::map_dbl(1:1000, function(i) {
  df <- dgp_broken(true_te = true_te)
  estimate_diff_in_means(df)
})

ggplot() +
  geom_histogram(aes(x = mc_dgp_broken), fill = "#20B2AA") +
  geom_vline(xintercept = true_te, linetype = "dashed") +
  labs(
    title = "DGP: `dgp_broken`",
    x = "Difference-in-means Estimate", y = "Count"
  ) +
  theme_bw(base_size = 12)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```