Trusting covariate adjustment

Alex Coppock

2023-10-26

Your're a reviewer for an experimental paper. It's not preregistered. The introduction and theory sections are first rate; the design section outlines an experiment that successfully randomizes the treatment of interest and seemingly measures the outcome of interest approximately without error. The units under study are agreed by all to be theoretically scoped-in. The experiment is a little small (just 500 units randomly assigned to treatment and control groups of exactly 250 units each) but it's a hard-to-study question. SUTVA is **not** a problem. Differential attrition is **not** a problem.

The results section presents two estimates of the average treatment effect, a difference-in-means estimate (DIM) and a covariate-adjusted estimate (OLS). The difference-in-means estimate is not statistically significant but the covariate-adjusted estimate is. In the abstract, introduction section, and conclusion section, the authors rely on the covariate-adjusted estimate and interpret their experiment as good evidence that the treatment affects the outcome, validating their theory.

Do you recommend that the editor publish the paper (i.e., offer suggestions for improvement that are unrelated to the core inference)? Or do you recommend that the paper be rejected (or amended such that the core inference is not made)? Suppose that if the DIM and OLS estimates were both significant, you would recommend publish for sure.

The right answer here is obvious: you recommend publish, because *IN THIS HOUSE* we condition publication decisions on design, not on results. Conditioning publication on statistical significance of either the DIM or the OLS estimator will generate bias in the empirical record. Moreover, if the covariates are correlated with the outome, the OLS estimator almost always has lower variance than the DIM estimator (see https://book.declaredesign.org/library/experimental-causal.html#can-controlling-for-covariates-hurt-precision for a discussion of exceptions), so if anything, we ought to prefer the OLS estimator over the DIM estimator.

But wait! We know that researchers can try out many covariate adjustment strategies. Since this experiment is not pre-registered, we can't be sure that the particular specification isn't the result of a p-hacking procedure. Maybe we recommend rejecting the study because we're worried the p-hacking estimator is *biased* and we want to use our gatekeeping powers as a reviewer to keep biased estimates out of the published record.

The rest of this short note will be dedicated to writing down the design (including the "p-hacking" answer strategy) in **MIDA** terms, which stands for Model, Inquiry, Data Strategy, Answer Strategy. M, I, D, A are the elements of a reseach design as described in our book on the topic; see https://book.declaredesign.org/introduction/what-is-a-research-design.html#mida-the-four-elements-of-a-research-design for a brief introduction to the MIDA framework).

The payoffs from this exercise will be twofold. First, we'll show that indeed p-hacking leads to bias; if you want to condition publication on unbiasedness, you need a mechanism (for example, preregistration) to be sure the specification wasn't chosen opportunistically. Second, we'll show that conditioning publication decisions on the pattern of statistical significance – even in the presence of p-hacking! – makes everything worse.

The design

The model is that all units have two potential outcomes, a treated and a control potential outcomes. These potential outcomes functions of an unobserved variable `U` and a treatment `Z`, which has a constant effect on all units of `effect_size`, set in this case to `0.1`. The unobserved variable `U` is correlated with two observed variables, `X1` and `X2` – all three are drawn from a multivariate normal distribution with a variance-covariance matrix specified as `vcov_mat`.

Important! If you want to tinker with this design, the place to start is by tinkering with the model to see how you conclusions about what to do as a reviewer depend on your beliefs about the model!

```
library(knitr) # for tables
library(tidyverse) # naturally
library(DeclareDesign)

N <- 500 # total subjects
r <- 0.5 # correlation for U with X1 and U with X2
effect_size = 0.1

vcov_mat <-
  matrix(c(1, r, r,
           r, 1, 0,
           r, 0, 1),
         nrow = 3, ncol = 3, byrow = TRUE)
model <-
  declare_model(N = N,
                draw_multivariate(
                  c(U, X1, X2) ~ MASS::mvrnorm(
                    n = N,
                    mu = c(0, 0, 0),
                    Sigma = vcov_mat
                  )),
                potential_outcomes(Y ~ effect_size * Z + U))
```

Next, we need to be specific about what this study is trying to learn: i.e., the inquiry. Here the inquiry is the average treatment effect, or the average difference between the treated and untreated outcomes. We built in a treatment effect of 0.1 for all units, so the inquiry is appropriately returning 0.1.

```
inquiry <- declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0))
design_so_far <- model + inquiry
run_design(design_so_far)
```

```
##   inquiry estimand
## 1     ATE      0.1
```

The data strategy for this study is to randomize exposure to the treatment for exactly half the units, then to measure post-treatment outcomes:

```
data_strategy <-
  declare_assignment(Z = complete_ra(N, m = 250)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z))
```

We'll entertain two answer strategies:

1. A difference-in-means estimator of the ATE
2. A "p-hacking" estimator that searches through four regression specifications and then presents the one that yield the lowest *p*-value on the ATE estimate

```
# this function runs the four regressions, then returns
# the estimate with the lowest p-value
p_hacker <-
  function(data) {
    fit_1 <- lm_robust(Y ~ Z + X1, data = data)
    fit_2 <- lm_robust(Y ~ Z + X2, data = data)
    fit_3 <- lm_robust(Y ~ Z + X1 + X2, data = data)
    fit_4 <- lm_robust(Y ~ Z + X1 * X2, data = data)

    lowest_p.value_estimate <-
    list(fit_1, fit_2, fit_3, fit_4) |>
      map_df(tidy) |>
      filter(term == "Z") |>
      arrange(p.value) |>
      slice(1)
  }

answer_strategy <-
  declare_estimator(Y ~ Z, .method = difference_in_means,
                    inquiry = "ATE", label = "DIM") +
  declare_estimator(handler = label_estimator(p_hacker),
                    inquiry = "ATE", label = "P-hacking")
```

Diagnosis

With all four elements in place, we can declare the design, then diagnose to calculate "diagnosands" or properties of the design: we'll consider bias, coverage, and rmse.

```
design <- model + inquiry + data_strategy + answer_strategy
diagnosis <- diagnose_design(design)

# this part is just making the table look nice-ish
diagnosis |>
  tidy() |>
  filter(diagnosand %in% c("bias", "coverage", "rmse")) |>
  select(estimator, diagnosand, estimate, std.error) |>
  arrange(diagnosand) |>
  kable()
```

estimator	diagnosand	estimate	std.error
DIM	bias	0.0034276	0.0039695
P-hacking	bias	0.0269306	0.0031434
DIM	coverage	0.9640000	0.0078041
P-hacking	coverage	0.9180000	0.0127091
DIM	rmse	0.0836702	0.0027196
P-hacking	rmse	0.0760753	0.0027091

As we always thought, the p-hacking estimator is bad! While the difference-in-means estimator is unbiased, the p-hacking estimator is upwardly biased. Coverage is below nominal for the p-hacking estimator. The one silver lining is that the rmse is lower for the p-hacking estimator, so depending on your view of the bias-variance tradeoff, you might make an argument for the p-hacking estimator on rmse grounds.

Publication decisions

Now let's consider what happens if you, as a reviewer, decide to accept or reject the paper on the basis of the pattern of statistical signifiance. Suppose you follow the following rule: If the DIM and (p-hacked) OLS are both statistically significant, you recommend publishing, but you recommend reject otherwise. Your reasoning goes like this: If only the DIM is significant but not the OLS, the result is "fragile." If only the OLS is significant but not the DIM, you suspect that the result is "p-hacked," so you recommend rejection. If neither estimate is significant, you recommend reject because the study is "underpowered."

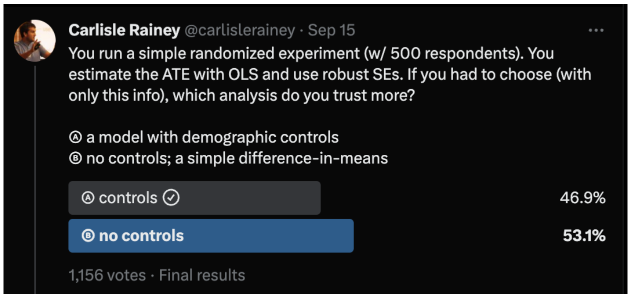
```r
library(reshape2) # for melt and dcast

# this pipeline reshapes the simulations from "long" to "wide"
sims_wide <-
  diagnosis |>
  get_simulations() |>
  select(sim_ID, estimator, estimand, estimate, p.value) |>
  melt(id.vars = c("estimator", "estimand", "sim_ID")) |>
  dcast(sim_ID + estimand ~ estimator + variable) |>
  mutate(
    significance =
      case_when(DIM_p.value <= 0.05 & `P-hacking_p.value` <= 0.05 ~ "Both significant",
                DIM_p.value <= 0.05 & `P-hacking_p.value` > 0.05 ~ "Only unadjusted",
                DIM_p.value > 0.05 & `P-hacking_p.value` <= 0.05 ~ "Only adjusted",
                DIM_p.value > 0.05 & `P-hacking_p.value` > 0.05 ~ "Neither significant")
  )

sims_wide |>
  group_by(significance) |>
  summarise(DIM_bias = mean(DIM_estimate - estimand),
            `P-hacking_bias` = mean(`P-hacking_estimate` - estimand)) |>
  kable()
```

significance	DIM_bias	P-hacking_bias
Both significant	0.1236144	0.1137500
Neither significant	-0.0448125	-0.0187546
Only adjusted	0.0194070	0.0652236
Only unadjusted	0.0929162	0.0403310

Here we see that bias in the "Both significant" row is quite bad for both estimators and is in fact worse for the DIM estimator than the p-hacked estimator!. Conditioning publication decisions on statistical significance is bad even if we suspect p-hacking **and we're right** that the estimate is p-hacked!

So what should we do? We should condition publication decisions on design, not results. If the design is not pre-registered, leaving the door open for unscrupulous specification searches, then we indeed might want to recommend rejection – but that decision has to be made independent of the pattern of statistical significance.

Last point: this post was inspired by this Twitter poll from Carlisle Rainey:



Carlisle Rainey @carlislerainey · Sep 15

You run a simple randomized experiment (w/ 500 respondents). You estimate the ATE with OLS and use robust SEs. If you had to choose (with only this info), which analysis do you trust more?

Ⓐ a model with demographic controls
Ⓑ no controls; a simple difference-in-means

Ⓐ controls ✓ 46.9%
Ⓑ no controls 53.1%

1,156 votes · Final results

In the poll, the question is much more straightforward: between DIM and OLS estimators, which do you trust more? Here there's no p-hacking, we can just characterise the bias and standard error of the two quite easily:

```
design <-
  model +
  inquiry +
  data_strategy +
  declare_estimator(Y ~ Z, .method = difference_in_means,
                    inquiry = "ATE", label = "DIM") +
  declare_estimator(Y ~ Z + X1 + X2, .method = lm_robust,
                    inquiry = "ATE", label = "OLS")

design |>
  diagnose_design() |>
  tidy() |>
  select(estimator, diagnosand, estimate, std.error) |>
  filter(diagnosand %in% c("bias", "sd_estimate")) |>
  arrange(diagnosand) |>
  kable()
```

estimator	diagnosand	estimate	std.error
DIM	bias	0.0039376	0.0037311
OLS	bias	0.0023427	0.0026626
DIM	sd_estimate	0.0896135	0.0027389
OLS	sd_estimate	0.0629145	0.0020697

Here we see the canonical result that OLS typically achieves a lower standard error than DIM with neglible bias, which is why I chose "controls" when I answered the poll.

Exercise for the reader

- What happens to the bias of the p-hacking estimator as the effect size gets bigger? As the sample size gets bigger?