The Statistical Crisis in Science

or: How I Learned to Stop Worrying and Love Insignificant Results

Brenton Kenkel — PSCI 8357
January 28, 2016

This week is ostensibly about making inferences from regression results. I assume by now you know the basics of testing hypotheses about OLS estimates: divide the coefficient estimate by its estimated standard error and then consult your Z tables (or t tables, if you're into the whole normality thing).

Instead of deriving the asymptotic variance of the OLS estimator, we're going to talk about real-world problems of inference. I will try to convince you:

- Reported effects in scientific publications are systematically overestimated.
- Reported p-values in scientific publications are systematically underestimated.
- The convention of only publishing significant results causes these problems. This practice is statistically ill-founded and encourages working scientists to engage in unethical research practices.
- You should judge your own work and others' on the basis of their research design, not whether they yield significant results.

The Statistical Significance Filter

If you open up an issue of any empirically oriented political science journal, you will not read many abstracts that conclude "We were unable to reject the null hypothesis of no effect." You probably won't see any. The prevailing attitude of reviewers and editors is that only significant results are interesting and only interesting results are worth publishing—so only significant results get published.

Consequently, published empirical findings are not a representative sample of all empirical findings. Andrew Gelman calls this the *statistical significance filter*: the publication process only reveals the findings of some studies, namely those

that achieve statistical significance. If you draw your beliefs from scientific journals (particularly prestigious ones, as Ioannidis (2008) notes), you will end up with some false ideas about how the world works.

Some of these beliefs will be Type I errors: you will reject null hypotheses that are true. Suppose there is a treatment T that has no effect on an outcome Y, and 100 labs run separate experiments of the effect of T on Y. We would expect about 95 of these experiments to (correctly) fail to reject the null hypotheses, and about 5 to (incorrectly) reject it. But if some of the significant findings get published and none of the insignificant ones do, you will end up incorrectly believing the treatment affects the outcome.

But the statistical significance filter has another, less obvious—and thus more pernicious—effect on our inferences. Assume that the null hypothesis is indeed false: that the treatment T has an effect on the outcome Y. Suppose once again that 100 labs run separate experiments of the effect of T on Y. Depending on the power of the experiments (a crucial point we will revisit in a minute), some proportion of them will (incorrectly) fail to reject the null hypothesis, and the remainder will (correctly) reject it. Because of the statistical significance filter, only the ones that reject the null hypothesis will get published.

That's not so bad, right? Only the studies that reject the null hypothesis get published, but the null hypothesis is wrong! The problem comes in when we want to evaluate the size of the effect—what political scientists like to call "substantive significance."[1] On average, the statistically significant studies will tend to overestimate the magnitude of the effect. Viewing studies through the statistical significance filter, we will correctly infer that there is an effect, but we will systematically overestimate how strong it is.

The first time I read about this result, on Andrew Gelman's blog, I didn't believe it. (I *should* have believed it, because he's a professional statistician and I'm not.) So I fired up R and ran a simulation to answer: if we only report our estimate of β_j when it's statistically significant, will we overestimate its magnitude on average? Today we'll run a version of that same simulation.

[1] Mini rant: In my admittedly short career in political science, I have seen zero talks or papers claim to have found a statistically significant but substantively insignificant result. I have, however, seen talks that claimed a 0.001% increase constituted a substantively significant finding. Without a threshold for substantive significance that is decided on *before* the results are obtained, any claim about substantive significance is incredible.

I'll begin by loading some useful packages. **dplyr**, **foreach**, and **ggplot2** are familiar by now. **broom** is a new one: we use it to "sweep" regression results into easy-to-use data frames.

```
library("dplyr")
library("foreach")
library("ggplot2")
library("broom")
```

I'm going to assume there is a binary treatment T, with a 50-50 chance of each observation being in the treatment or control group. The response Y is a function of the treatment and random error,

$$Y = \alpha + \beta T + \epsilon,$$

where $\epsilon \sim N(0, 1)$.

Assume the sample size is $N = 200$ and the true parameters are $\alpha = 1$ and $\beta = 0.1$. Let's take 100 draws from the sampling distribution of the OLS estimator of β and its associated p-value, one for each hypothetical lab running this hypothetical experiment, each with a distinct set of 200 subjects. Remember that we use the `replicate()` function to run the same operation repeatedly.

```
n_obs <- 200
alpha <- 1
beta <- 0.1

beta_hat_dist <- replicate(100, {
    ## Simulate data
    treatment <- rbinom(n_obs, 1, 0.5)
    response <- alpha + beta * treatment + rnorm(n_obs)

    ## Fit regression model and extract estimates related to the
    ## treatment variable
    ols_fit <- lm(response ~ treatment)
    ols_coef <- tidy(ols_fit) %>% filter(term == "treatment")

    ## Return the coefficient and the p-value on treatment
    c(beta_hat = ols_coef$estimate,
      p_value = ols_coef$p.value)
})
```

`replicate()` returns the result of each iteration in a separate column, so let's take its transpose and turn it into a data frame.

```
dim(beta_hat_dist)
```

```
## [1]   2 100
```

```
beta_hat_dist <- as.data.frame(t(beta_hat_dist))
head(beta_hat_dist)
```

```
##   beta_hat p_value
## 1   -0.029  0.8282
## 2    0.134  0.3206
## 3    0.207  0.1408
## 4    0.278  0.0313
## 5    0.139  0.2968
## 6   -0.218  0.1247
```

The OLS estimator is unbiased, so if we take the mean of our draws from the sampling distribution, they should roughly equal the true value, $\beta = 0.1$.

```
beta_hat_dist %>%
    summarise(e_beta_hat = mean(beta_hat))
```

```
##   e_beta_hat
## 1      0.092
```

Yep. But now what if we split up the results by significant and insignificant? Does the expected value of $\hat{\beta}$, conditional on it being statistically significant, still equal the true value?

```
beta_hat_dist <- mutate(beta_hat_dist,
                        significant = p_value <= 0.05)

beta_hat_dist %>%
    group_by(significant) %>%
    summarise(count = n(),
              e_beta_hat = mean(beta_hat))
```

```
## Source: local data frame [2 x 3]
##
##   significant count e_beta_hat
## 1       FALSE    89     0.0675
```

```
## 2        TRUE    11     0.2895
```

Two things to notice here. First, the average estimate conditional on statistical significance is overstated—about triple the true value. Second, the power is pretty low. The null hypothesis is false, but we fail to reject it about 90% of the time.

These two phenomena are related. Low power means we rarely reject the null hypothesis—only for the most extreme estimates of the effect. So if we only observe the estimate conditional on it being statistically significant, we're only seeing draws from the tail of the sampling distribution.

To see how power affects the bias induced by the statistical significance filter, let's run the same simulation for different values of β. The stronger the effect of the treatment, the higher the signal-to-noise ratio and thus the greater the power of the study. Let's run this simulation for each $\beta \in \{0.2, 0.3, 0.4, 0.5\}$.

```
beta_seq <- seq(0.2, 0.5, by = 0.1)
n_obs <- 200
alpha <- 1

foreach (beta = beta_seq) %do% {
    beta_hat_dist <- replicate(100, {
        ## Simulate data
        treatment <- rbinom(n_obs, 1, 0.5)
        response <- alpha + beta * treatment + rnorm(n_obs)

        ## Fit regression model and extract estimates related to the
        ## treatment variable
        ols_fit <- lm(response ~ treatment)
        ols_coef <- tidy(ols_fit) %>% filter(term == "treatment")

        ## Return the coefficient and the p-value on treatment
        c(beta_hat = ols_coef$estimate,
          p_value = ols_coef$p.value)
    })

    beta_hat_dist <- as.data.frame(t(beta_hat_dist))

    beta_hat_dist <- mutate(beta_hat_dist,
                            significant = p_value <= 0.05)
```

```
    beta_hat_dist %>%
        group_by(significant) %>%
        summarise(count = n(),
                  e_beta_hat = mean(beta_hat))
}
```

```
## [[1]]
## Source: local data frame [2 x 3]
##
##   significant count e_beta_hat
##         (lgl) (int)      (dbl)
## 1       FALSE    71      0.131
## 2        TRUE    29      0.346
##
## [[2]]
## Source: local data frame [2 x 3]
##
##   significant count e_beta_hat
##         (lgl) (int)      (dbl)
## 1       FALSE    51      0.179
## 2        TRUE    49      0.402
##
## [[3]]
## Source: local data frame [2 x 3]
##
##   significant count e_beta_hat
##         (lgl) (int)      (dbl)
## 1       FALSE    16      0.211
## 2        TRUE    84      0.454
##
## [[4]]
## Source: local data frame [2 x 3]
##
##   significant count e_beta_hat
##         (lgl) (int)      (dbl)
## 1       FALSE     7      0.218
## 2        TRUE    93      0.519
```

The bigger the true effect, the greater the power of the hypothesis test, and

thus the more the sampling distribution conditional on statistical significance comes to resemble the full sampling distribution. Unfortunately, the typical setting in political science is the one where the statistical significance filter is most severe. Nonzero but small relationships among variables are common in political science, as are small sample sizes.

So the magnitudes of the estimates we see—the ones that make it past the statistical significance filter—are biased toward overestimating effects. What can we do about it?

- Assume the magnitudes of published results are exaggerated, and adjust our own beliefs accordingly.
- Collect new data to replicate published findings, and adjust our beliefs in the direction of the replication results.
- When writing our own papers, **don't** throw results away just because they're "insignificant."
- When reviewing others' papers, **don't** judge on the basis of significance. Try to be "results-blind." Assess whether the research design is well suited to address the question at hand, not whether it turned up the results the author wanted, or the results you want, or interesting or surprising or counterintuitive results, etc.

p-Hacking

The statistical significance filter is a demand-side problem. The demand (by journals) for "insignificant" findings is too low. This in turn creates supply-side problems. Scientists' careers depend on their ability to publish their findings. Since there is no demand for insignificant findings, scientists do what they can to conjure up significant results. In the best case scenario, this means devoting effort to projects with a high prior probability of turning up significant, rather than riskier endeavors. In the worst case, it means engaging in vaguely-to-definitely unethical statistical practices in a desperate search for significance.

Let us once again imagine a lab performing an experiment. They are interested in the effect of a treatment T on an outcome Y. To make it concrete, suppose the treatment is reading a particular editorial, and the outcome is where the respondent places himself or herself on a left-right ideological scale ranging between 0 and 1. The lab spends a lot of time and money recruiting subjects,

running the experiment, and tabulating the data. They get their spreadsheet together, load their data into R, test for a treatment effect ... and fail to reject the null hypothesis.

Damn. All that effort wasted, for a result that can't be published. But wait! The op-ed was written by a man, and his picture appeared next to it. It seems plausible that it might only have an effect on men, or only one on women. So just to see, the lab re-runs the test once just for men and once just for women. They get a p-value just below 0.05 for the male subsample! Hooray! This is at least potentially a publishable finding!

What's wrong with this picture? Let's go back to the formal definition of the significance level.

> The significance level of a hypothesis test is the probability of rejecting the null hypothesis when the null hypothesis is true.

If the null hypothesis is true, and 100 labs run the same experiment on it, we should expect about 5 of them to end up incorrectly rejecting the null hypothesis. Similarly, go back to the formal definition of a p-value.

> The p-value of a test statistic is the probability of yielding a test statistic at least as extreme when the null hypothesis is true.

If the null hypothesis is true, we should expect only about 10 out of 100 labs to end up with $p \leq 0.10$, 5 out of 100 to have $p \leq 0.05$, and so on.

The problem with this hypothetical procedure—testing *post hoc* for effects within subgroups after the main test comes back insignificant—is that the stated significance level is not the real significance level. If you run three different tests and reject the null hypothesis if *any* of them comes back with $p \leq 0.05$, you will reject the null hypothesis more often than 5% of the time. In our running hypothetical example, the lab's reported p-value of 0.05 is a lie.

Let's run a simulation to see exactly how often we reject a true null hypothesis, assuming the response $Y \sim U[0,1]$ is independent of the binary treatment T and we analyze the hypothesis on the full sample and two subsamples. To begin, let's simulate a single set of data and run a regression on it.

```
n_obs <- 100
treatment <- rbinom(n_obs, 1, 0.5)
male <- rbinom(n_obs, 1, 0.5)
```

```
response <- runif(n_obs)

fit_all <- lm(response ~ treatment)
summary(fit_all)
```

```
##
## Call:
## lm(formula = response ~ treatment)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5017 -0.2482 -0.0257  0.2493  0.5537
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5052     0.0420   12.03   <2e-16
## treatment    -0.0648     0.0577   -1.12     0.26
##
## Residual standard error: 0.288 on 98 degrees of freedom
## Multiple R-squared:  0.0127, Adjusted R-squared:  0.00263
## F-statistic: 1.26 on 1 and 98 DF,  p-value: 0.264
```

We want to run two more regressions like this, one for each gender subgroup, and extract the p-value on the treatment variable from each of them. We'll write a function to perform this extraction.

```
extract_p <- function(fitted_model) {
    tidy(fitted_model) %>%
        filter(term == "treatment") %>%
        select(p.value) %>%
        as.numeric()
}

extract_p(fit_all)
```

```
## [1] 0.264
```

Now we can run the regressions on each subgroup and get their p-values too.

```
fit_male <- update(fit_all, subset = male == 1)
fit_female <- update(fit_all, subset = male == 0)
```

```
extract_p(fit_male)
```

```
## [1] 0.894
```

```
extract_p(fit_female)
```

```
## [1] 0.0194
```

OK, so now we've done it once. But we want to repeat it many times, to see what percentage of the time we end up with at least one p-value below 0.05. To do this we'll turn to our friend `replicate()`.

```
n_obs <- 100

sim_p_hack <- replicate(1000, {
    ## Simulate data
    treatment <- rbinom(n_obs, 1, 0.5)
    male <- rbinom(n_obs, 1, 0.5)
    response <- runif(n_obs)

    ## Run regressions
    fit_all <- lm(response ~ treatment)
    fit_male <- update(fit_all, subset = male == 1)
    fit_female <- update(fit_all, subset = male == 0)

    ## Extract p-values
    p_all <- extract_p(fit_all)
    p_male <- extract_p(fit_male)
    p_female <- extract_p(fit_female)

    ## Report the lowest p-value
    min(p_all, p_male, p_female)
})
```

Let's see what proportion of the time we end up with a reported p-value less than 0.05. Remember, since the null hypothesis is true, this should be 5%.

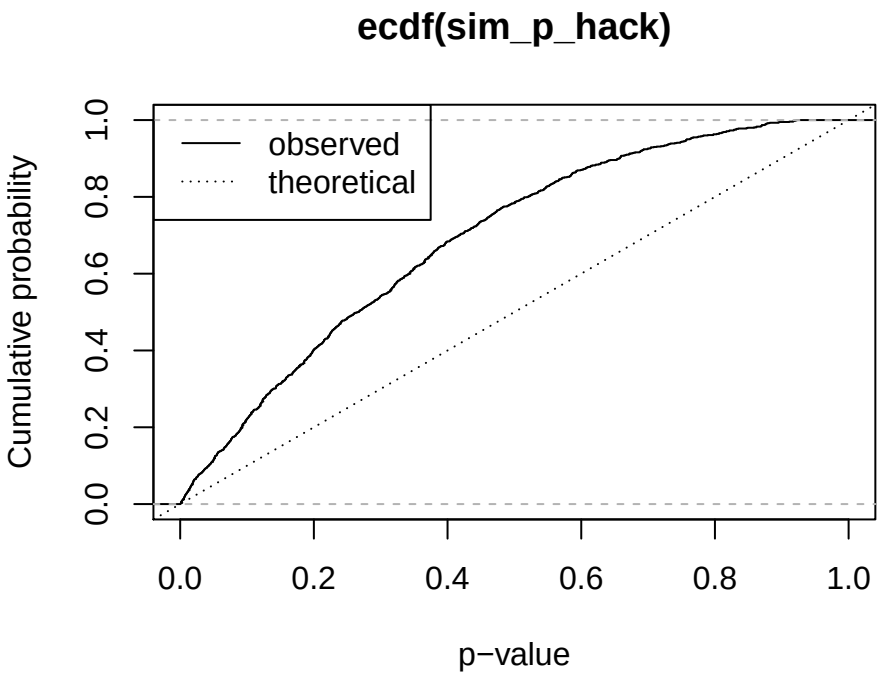
```
mean(sim_p_hack <= 0.05)
```

```
## [1] 0.115
```

The actual probability of incorrectly rejecting the null hypothesis is more than double what it should be. And that's just with a single *post hoc* subgroup split. With enough *post hoc* slicing and dicing of the data, the probability of finding some nominal $p \leq 0.05$ can get pretty big.

To drive the point home further, let's look at the distribution of the reported p-values under the null hypothesis.

```
plot(ecdf(sim_p_hack),
     xlab = "p-value",
     ylab = "Cumulative probability",
     xlim = c(0, 1))
abline(a = 0, b = 1, lty = 3)
legend("topleft", c("observed", "theoretical"), lty = c(1, 3))
```



We've talked about one method of p-hacking, but there are many ways:

- Splitting up data by subgroups *post hoc*
- Changing the set of variables you control for
- Changing the operationalization of the covariate of interest or the response variable
- Changing the time period of the analysis

- Stopping data collection as soon as $p \leq 0.05$

What all these have in common is that the final test you report depends on the result of some earlier test you ran. All standard hypothesis tests assume that you didn't do anything like this—that this was the only test you ran, that your initial results didn't influence your choice of further tests. It is unethical to report the nominal p-value (i.e., the value your computer spits out) from a p-hacked test, because the true probability of getting a result at least as extreme is greater than the nominal value.

What should you do to get by in political science while maintaining high ethical standards?

- Decide exactly which hypothesis you want to test and which test to run before you collect your data, or at least before running any analysis on it.
- Report every test you perform on the data, and only highlight results that are robust across tests.
- Randomly split your sample before performing any tests. Go wild with the first half of the sample looking for an interesting hypothesis. Then test that hypothesis on the other half of the sample (and report the results whether they come out in your favor or not). Equivalently, hack your pilot data and then go out and collect new data to try to replicate your hacked initial hypothesis.
- Apply a correction for multiple testing problems, or use computational methods to calculate the distribution of a data-conditional test statistic under the null hypothesis. (We'll talk a bit about how to do this in the latter weeks of the course.)
- Give up on all this data stuff and become a formal theorist.

References

Ioannidis, John P. A. 2008. "Why Most Discovered True Associations Are Inflated." *Epidemiology* 19 (5): 640–48. doi:10.1097/EDE.0b013e31818131e7.