CODERS' CORNER

CENTRE FOR THE STUDY OF AFRICAN ECONOMIES

SIMULATION-BASED POWER CALCULATIONS

If you want to run power calculations using your baseline data, you can (typically) either use the asymptotic properties of your sample **or** bootstrap your βi using simulations on your baseline data. One advantage of using simulations is that you can add controls to your power analysis. Typically, policies/interventions only mildly affect the variance of the outcome of interest. So even if your policy has a (relatively) large impact, the standard errors (SEs) under the null hypothesis can be similar to the SE under the alternative. Adding controls can help get more precise measures of the impact of interest.

In this post, I will walk you through how to run these simulations, using a command attached to this post.

Syntax

```
powersimz outcome_var, covariates() [iterations() seed() level() treatmentshare() takeup() alpha() power() ]
```

Description

powersimz calculates and stores what I call theoretical and empirical simulation-based minimum detectable effects (MDEs).

i) Theoretical MDEs: these MDEs are based on the SEs returned by the regress command, which rely on modeling assumptions (e.g. normally distributed errors in the case of OLS, where the SE is just $\sqrt{\sigma(X'X)^{-1}}$.) In this case the MDE is essentially: $(t_{\alpha/2} + t_{1-k}) * Mean(SEs)$ of all the SEs from the β on each of the simulated treatment.

ii) Empirical MDEs: these MDEs are permutation-based, so there are no modeling assumptions. To generate the MDEs we take SD of the permutation sample of β (i.e. the sample of beta-hats from the simulations). The empirical version is based on the logic of permutation tests, which shuffle the treatment vector in order to simulate the null distribution. In this case the MDE is essentially: $(t_{\frac{\alpha}{2}} + t_{1-k}) * SD(\hat{\beta})$

Note that MDEs are defined by the following: $(t_{\frac{\alpha}{2}} + t_{1-k}) * \sigma(\hat{\beta})$

The main difference between the two is how we define σ($\hat{\beta}$): in the "theoretical" version I am approximating σ($\hat{\beta}$) with the mean of the standard errors of the $\hat{\beta}$ from all the simulated randomizations (this makes the underlying assumptions that errors are normally distributed), and in the "empirical" version I approximate σ($\hat{\beta}$) with the standard deviation of all the $\hat{\beta}$ from the same simulations (this makes less stringent modeling assumptions).

Downloading powersimz

Step 1: Download the .zip file attached here [embed link], extract the contents in your local computer

Step 2: Open Stata, and type the following:

```
adopath + "~/Downloads/powersimz"

help powersimz
```

Step 3: Create a temp global

```
global temp "[path]"
```

Example:

```
global temp "~/Documents/temporary" // replace this with the path to whatever folder you want to make your temporary folder


clear
set obs  1200


egen schools = fill(1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21)


sort schools


gen x1 = rnormal(0 , 1)
gen x2 = rbinomial(1200 ,  .5)
gen y   = rnormal(4 , 10)

************************** Iterations


/* 1 */ powersimz y, cov(x2 x1 schools) level(schools) iter(100)
return list


/* 2 */ powersimz y, cov(x2 x1 schools) level(schools) iter(200)
return list


/* 3 */ powersimz y, cov(x2 x1 schools) level(schools) iter(500)
return list


/* 4 */ powersimz y, cov(x2 x1 schools) level(schools) iter(1000)
return list
```

Note that the more iterations you set as an option, the more reliable the MDEs, especially for the permutation-based ones. Whether you choose the permutation based (empirical) MDE or the one that assumes normality of the error term (theoretical) depends on what you believe your $\sigma(\hat{\beta})$ is. Note that for the theoretical MDE (asymptotic assumptions), the variation in the MDEs across the different outputs of the command should be small and due to: a) variation in $\hat{\sigma}$, which is expected to be small; that is because the only difference across the simulations is the treatment vector which is by design independent from the outcome y, and b) the fact that the correlation between the treatment vector and the other predictors isn't exactly zero.

```
************************** Take-up


/* 5 */ powersimz y, cov(x2 x1) level(schools) iter(100) seed(20190630)
return list


/* 6 */ powersimz y, cov(x2 x1) level(schools) iter(100) takeup(.60) seed(20190630)
return list
```

Note that the MDE decreases with the take up rate. In /* 6 */ I assume that 60 % of the individuals who are offered the treatment actually take it up. Compared with /* 5 */, where there is a 100% take-up, you can see the change in MDE resulting from that drop in hypothetical treatment take-up.

```
************************** Treatment share


/* 7 */ powersimz y, cov(x2 x1) level(schools) iter(100)  seed(20190630)
return list


/* 8 */ powersimz y, cov(x2 x1) level(schools) iter(100) treatmentshare(.2) seed(20190630)
return list
```

Here we play with the share of individuals in the sample who are treated. Say that you have a base-line survey or an administrative dataset on a large population but you can only afford to treat 20% of them (as opposed to the typical 50%). You can see the difference by comparing the output of /* 7 */ and /* 8 */.

Binta Zahra Diop, DPhil Candidate in Economics, St Hugh's College, Oxford
29 May 2019