version 0.1.0

Title

persuasio — Conduct causal inference on persuasive effects

Syntax

persuasio *subcommand varlist* [*if*] [*in*] [, **level**(#) **model**(*string*) **method**(*string*) **nboot**(#) **title**(*string*)]

Options

Option	*Description*
level(#)	Set confidence level; default is **level**(95)
model(*string*)	Regression model when *covariates* are present
method(*string*)	Inference method; default is **method**("normal")
nboot(#)	Perform # bootstrap replications
title(*string*)	Title of estimation

Description

persuasio conducts causal inference on persuasive effects. It is a wrapper that calls a variety of subroutines. *subcommand* has several options:

Subcommand	*Description*
apr	inference on APR when y,t,z are observed
lpr	inference on LPR when y,t,z are observed
yz	inference on APR and LPR when y,z are observed
calc	bound estimates on APR and LPR with summary statistics

apr and **lpr** refer to a data scenario where binary outcomes *y*, binary treatments *t*, and binary instruments *z* are observed (with covariates *x* if exist) for each observational unit. **apr** and **lpr** provide causal inference on the average persuasion rate (APR) and the local persuasion rate (LPR), respectively.

yz is concerned with another data scenario where persuasive treatment *t* is unobserved. In this case, bounds on the APR are the same as those on the LPR. It provides causal inference for the APR and hence, for the LPR as well.

calc is designed for the case when summary statistics on Pr(y=1|z) and/or Pr(t=1|z) for each z=0,1 are available. It provides the lower and upper bounds on the APR as well as the lower and upper bounds on the LPR.

Options

model(*string*) specifies a regression model of *y* on *z* and *x*.

This option is only relevant when *x* is present. The default option is "no_interaction" between *z* and *x*. When "interaction" is selected, full interactions between *z* and *x* are allowed.

level(#) sets confidence level; default is **level**(95).

method(*string*) refers the method for inference.

The default option is **method**("normal"). By the naure of identification, one-sided confidence intervals are produced.

1. When *x* is present, it needs to be set as **method**("bootstrap"); otherwise, the confidence interval will be missing.

2. When *x* is absent, both options yield non-missing confidence intervals.

nboot(#) chooses the number of bootstrap replications.

The default option is **nboot**(50). It is only relevant when **method**("bootstrap") is selected.

title(*string*) specifies the title of estimation.

All these options are irrelevant for subcommands **calc**.

Remarks

It is recommended to use **nboot**(#) with # at least 1000. A default choice of 50 is meant to check the code initially because it may take a long time to run the bootstrap part. The bootstrap confidence interval is based on percentile bootstrap. A use of normality-based bootstrap confidence interval is not recommended because bootstrap standard errors can be unreasonably large in applications.

Examples

We first call the dataset included in the package.

```
. use GKB, clear
```

The first example conducts inference on APR when y,t,z are observed.

```
. persuasio apr voteddem_all readsome post, level(80) method("normal")
```

The second example conducts inference on LPR when y,t,z are observed.

```
. persuasio lpr voteddem_all readsome post, level(80) method("normal")
```

The third example conducts bootstrap iinference on APR and LPR when y,z are observed with a covariate, MZwave2, interacting with the instrument, post.

```
. persuasio yz voteddem_all post MZwave2, level(80)
model("interaction") method("bootstrap") nboot(1000)
```

The fourth example considers the case when we have summary statistics on Pr(y=1|z) and/or Pr(t=1|z).

We first compute summary statistics.

```
  . foreach var in voteddem_all readsome {
      foreach treat in 0 1 {
        sum `var' if post == `treat'
        scalar `var'_treat' = r(mean)
      }
    }
```

Then, we calculate the bound estimates on the APR and LPR.

```
. persuasio calc voteddem_all_1 voteddem_all_0 readsome_1 readsome_0
```

Stored results

apr calls this package's command **persuasio4ytz**, **lpr** command **persuasio4ytz2lpr**, **yz** command **persuasio4yz**, and **calc** command **calc4persuasio**, respectively. Check help files for these commands for details on stored results.

Authors

Sung Jae Jun, Penn State University, <sjun@psu.edu>

Sokbae Lee, Columbia University, <sl3841@columbia.edu>

License

References

Sung Jae Jun and Sokbae Lee (2019), Identifying the Effect of Persuasion, arXiv:1812.02276 [econ.EM]