

CODERS' CORNER

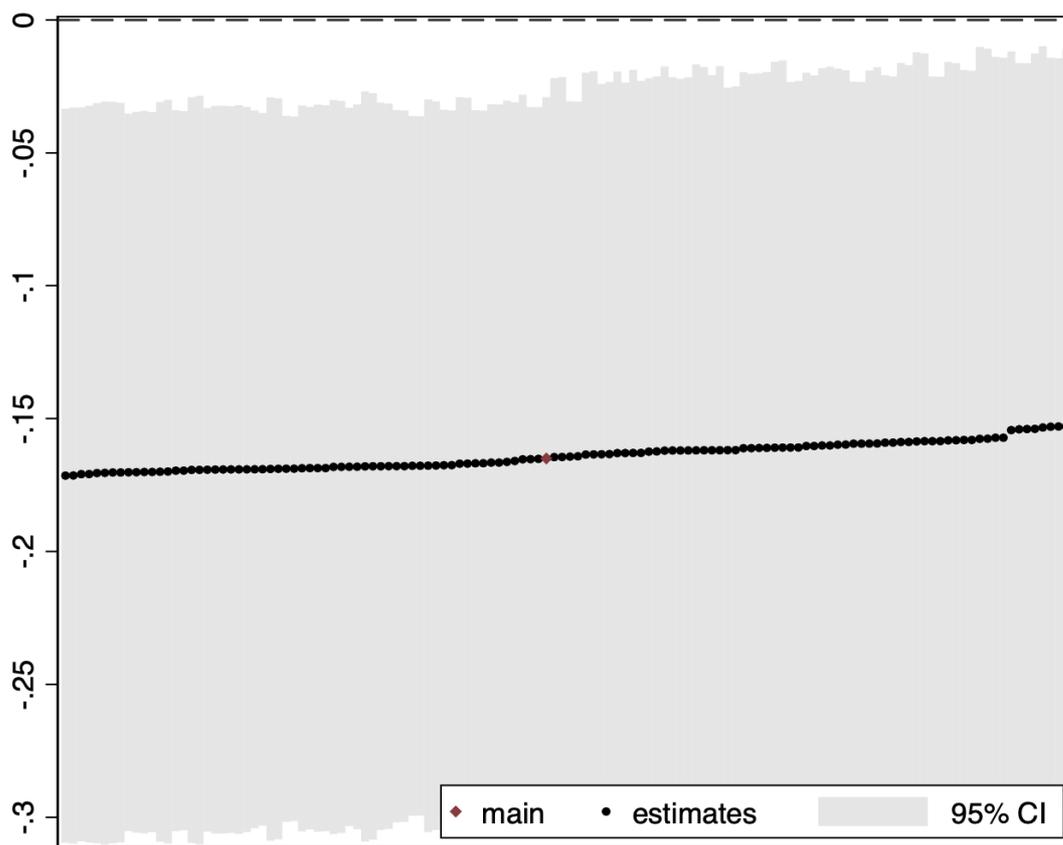


MIND & BEHAVIOUR
RESEARCH GROUP

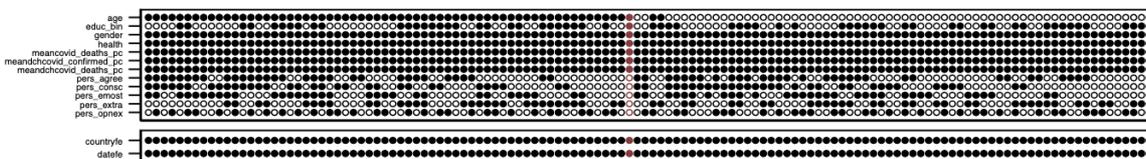


SPECIFICATION PLOTS USING SPECCURVE

Extensive robustness checks have become a requirement for empirical research. This often leads to Online Appendices with hundreds of result tables that are very hard to digest for readers and referees. Stata16's *speccurve* command written by [Martin Eckhoff Andresen](#) is an easy to use command that facilitates the generation of specification curves. The command can be download from [here](#). Unfortunately, the command does not currently work with older versions of Stata. A specification curve plots a large number of regression coefficients and confidence intervals sorted by estimated impact from different specifications that allow the assessment of robustness in a single figure like the one included below.



control variables



Creating specification curves requires two basic steps.

First, you need to run and save all regressions you would like to be included in the specification curve. These can be different sets of control variables. You will likely want to use loops to run and store a large number of different specifications. In the example do-file we use a loop to run a regression with all combinations of a set of seven different control variables. To construct the loop we use the `tokenize` command to store variables in locals named as numbers (eg in the below code the local `1' will have the string value "age").

```
tokenize age educ_bin pers_extra pers_agree pers_consc pers_emost pers_opnex
```

We then use a stacked loop to create a local that contains all control variables for a particular specification. In the loop we run the specification and store the regression results. Importantly, all variables including the treatment variable need to have the same name in all specifications (important if you would like to test for robustness with different treatment definitions). If we want to use the `panel()` option of `speccurve`, we also need to store additional variables in the regression results. Below we present a simplified code snippet that does exactly this.

```
loc no=0
forvalues age=0/1 {
    forvalues educ_bin=0/1 {
        forvalues pers_extra=0/1 {
            forvalues pers_agree=0/1 {
                forvalues pers_consc=0/1 {
                    forvalues pers_emost=0/1 {
                        forvalues pers_opnex=0/1 {
                            ** Create local with string of control variable names in specific specification **
                            loc controls
                            forvalues i=1/7 {
                                if ``i'==1 loc controls `controls' `i'
                            }
                            reg yvar lockdown `controls', robust
                            estadd scalar countryfe=0 // No fixed effect used
                            eststo ols`no' // Store regression results
                            loc ++no
                        }
                    }
                }
            }
        }
    }
}
```

Fixed effects can either be included as additional control variables or using additional loops and commands like `reghdfe` (which is what we do in the example do-file).

Second, you can use the `speccurve` command once all regression results are stored. We provide a brief overview over the syntax below (a more detailed version can be found in the excellent help file).

```
speccurve *, param(lockdown) controls(title(control variables)) main(ols97) panel(countryfe)
level(95)
```

`Speccurve` uses regression results stored in Stata's memory or in an external `.ster` file (the latter can be useful if running your model takes a long time). Here we use all specifications stored in the memory.

The options we use are:

- `param(lockdown)`
This specifies the coefficient estimate of interest (in our case whether a country imposed a COVID-19 related lockdown)
- `controls(title(control variables))`
The controls() option specifies how the title and labels for the control panel. Here we just specify the title.
- `main(ols97)`
The main() option specifies which specification has the preferred coefficient estimate and will be coloured in red (here specification 97).
- `panel(countryfe)`
The panel() option allows the displaying of differences in specifications not captured by control variables. These can be fixed effects but also things like different outcome and treatment definitions. Importantly, these differences have to be stored as binary scalar in the regression results.
- `level(95)`
The level() option specifies the desired confidence interval level.

Other options include `sort()` which specifies how coefficient estimates are sorted (default is by estimate size) and the usual graph options.

The example do-file contains code creating the included specification curve (Figure S4) from the paper “Perceptions of an Insufficient Government Response at the Onset of the COVID-19 Pandemic are Associated with Lower Mental Well-Being”. We make use of the data made publicly available by the authors of “Behaviors and Perceptions in the COVID-19 Pandemic” through the Open Science Foundation at <https://osf.io/3sn2k/>.