cfprobit — Control-function probit regression[+]

[+]This command is part of StataNow.

Description

`cfprobit` fits probit models for binary dependent variables with endogenous regressors using control functions. Endogenous variables are first modeled as a function of instruments using linear, probit, fractional probit, or Poisson regression. The residuals, or generalized residuals, from these first-stage regressions are then included in the main equation as control functions to make regression estimates robust to endogeneity.

Quick start

Control function estimates of a probit regression of binary variable `y1` on `x` and endogenous regressor `y2` that is instrumented by `z`

```
cfprobit y1 x (y2 = z)
```

Same as above, but with two endogenous regressors and two instruments

```
cfprobit y1 x (y2 y3 = z1 z2)
```

Same as above, but use `z3` as an additional instrument for `y3`

```
cfprobit y1 x (y2 = z1 z2) (y3 = z1 z2 z3)
```

Model the first stage for count variable `y4` using Poisson regression

```
cfprobit y1 x (y2 = z1 z2) (y4 = z1 z2 z3, poisson)
```

Include an interaction term between `w` and the control function of `y2` in the main equation

```
cfprobit y1 x (y2 = z, interact(w))
```

Include an interaction term between the control functions of `y2` and `y3`

```
cfprobit y1 x (y2 = z1 z2) (y3 = z1 z2 z3), cfinteract
```

Include `w` in the main equation for `y1` but not in the first stage

```
cfprobit y1 x (y2 = z), mainonly(w)
```

Include an endogenous interaction term between `w` and `y2`, and control for its endogeneity by including an interaction term between `w` and the control function of `y2`

```
cfprobit y1 x w (y2 = z, interact(w)), mainonly(c.y2#c.w)
```

Menu

Statistics > Endogenous covariates > Control-function probit regression

Syntax

```
cfprobit depvar [indepvars] (varlist_en1 = varlist_iv1 [, cfopts])
    [(varlist_en2 = varlist_iv2 [, cfopts]) ...] [if] [in] [weight] [, options]
```

cfopts	Description
Model	
`linear`	model the endogenous variables using linear regression; the default
`probit`	model the endogenous variables using probit regression
`fprobit`	model the endogenous variables using fractional probit regression
`poisson`	model the endogenous variables using Poisson regression
`interact(`*varlist*$_{int}$`)`	interact the variables in *varlist*$_{int}$ with the control functions

Only one of `linear`, `probit`, `fprobit`, or `poisson` is allowed in each set of parentheses.

options	Description
Model	
`mainonly(`*varlist*$_{m}$`)`	include the variables in *varlist*$_{m}$ as exogenous variables in the main equation but not in the first-stage equations
`cfinteract`	include interactions between control functions when there are multiple endogenous variables
`noconstant`	suppress constant term
`asis`	retain perfect predictor variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `robust`, `cluster` *clustvar*, `bootstrap`, `jackknife`, or `hac` *hacspec*
Reporting	
`level`	set confidence level; default is `level(95)`
`first`	report first-stage regressions
`noheader`	display only the coefficient table
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process
`coeflegend`	display legend instead of statistics

indepvars, *varlist*$_{en.}$, *varlist*$_{iv.}$, *varlist*$_{int}$, and *varlist*$_{m}$ may contain factor variables; see **[U] 11.4.3 Factor variables**.

depvars, *indepvars*, *varlist*$_{en.}$, *varlist*$_{iv.}$, *varlist*$_{int}$, and *varlist*$_{m}$ may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

`bootstrap`, `by`, `collect`, `jackknife`, `rolling`, and `statsby` are allowed; see **[U] 11.1.10 Prefix commands**.

Weights are not allowed with the `bootstrap` prefix; see [R] **bootstrap**.

`fweights`, `iweights`, and `pweights` are allowed; see **[U] 11.1.6 weight**.

`coeflegend` does not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

`linear`, `probit`, `fprobit`, and `poisson` specify which regression model is used for the first-stage model. A different model can be specified for each set of parentheses.

`linear`, the default, specifies a linear regression model.

`probit` specifies a probit regression model. Endogenous variables must be coded as 0/1.

`fprobit` specifies a fractional probit regression model. Endogenous variables must take values in $[0, 1]$.

`poisson` specifies a Poisson regression model. Endogenous variables must take nonnegative values.

`interact(`*varlist*$_{\text{int}}$`)` includes in the main regression an interaction term between each variable in *varlist*$_{\text{int}}$ and the control functions associated with the current set of parentheses. Variables are treated as continuous by default.

`mainonly(`*varlist*$_{\text{m}}$`)` includes the variables in *varlist*$_{\text{m}}$ as exogenous variables in the main regression but excludes them from the first-stage regressions.

`cfinteract` specifies that all interactions between control functions be included in the main regression. If there is only one endogenous regressor, and thus only one control function, the option has no effect.

`noconstant`; see [R] **Estimation options**.

`asis` requests that all specified variables and observations be retained in the maximization process. This option is typically not used and may introduce numerical instability. Normally, `cfprobit` omits any endogenous or exogenous variables that perfectly predict success or failure in the dependent variable. The associated observations are also excluded. For more information, see *Model identification* in [R] **probit**.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] ***vce_option***.

`vce(conventional)`, the default, requests conventional standard errors appropriate under homoskedasticity and no autocorrelation.

`vce(hac` *hacspec*`)` requests a heteroskedasticity- and autocorrelation-consistent (HAC) variance–covariance matrix. The full syntax of *hacspec* is one of the following:

`vce(hac` *kernel* `[` #`])` requests a HAC variance–covariance matrix using the specified kernel (see below) with optional # lags. The bandwidth of a kernel is equal to # + 1. If # is not specified, a kernel with $N - 2$ lags is used, where N is the sample size.

`vce(hac` *kernel* `opt [` #`])` requests a HAC variance–covariance matrix using the specified kernel (see below), and the lag order is selected using Newey and West's (1994) optimal lag-selection algorithm. # is an optional tuning parameter that affects the lag order selected; see the discussion in *Methods and formulas* in [R] **ivregress**.

kernel may be one of the following:

bartlett or nwest requests the Bartlett (Newey–West) kernel.

parzen or gallant requests the Parzen (Gallant 1987) kernel.

quadraticspectral or andrews requests the quadratic spectral (Andrews 1991) kernel.

Reporting

level(#); see [R] **Estimation options**.

first requests that the results of first-stage regressions be displayed.

noheader suppresses the display of the summary statistics at the top of the output, displaying only the coefficient table.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance; see [R] **Maximize**. These options are seldom used.

The following option is available with cfprobit but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

stata.com

Remarks and examples

cfprobit fits probit models for binary dependent variables with endogenous regressors by estimating one or more control functions and including them in the main regression equation. These control functions are estimated as the residuals, or generalized residuals, of first-stage regressions.

Control-function methods make use of instruments and are thus related to standard instrumental-variables methods. However, control-function methods allow for more flexibility than comparable instrumental-variables methods. Wooldridge (2015) gives an overview of control-function regression methods.

cfprobit fits a model whose main equation has the form

$$P(y_{i0} = 1|\mathbf{y}_i, \mathbf{x}_i, \mathbf{w}_i) = \Phi(\mathbf{y}_i\boldsymbol{\beta}_1 + \mathbf{x}_i\boldsymbol{\beta}_2 + \mathbf{w}_i\boldsymbol{\beta}_3 + u_i)$$

where y_{i0} is the dependent variable for the ith observation; $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{ip})'$ is a row vector of p endogenous regressors; $\mathbf{x}_i$ is a row vector of exogenous regressors to be included in the main equation and in first-stage regressions; $\mathbf{w}_i$ is a row vector of exogenous regressors to be included only in the main equation; $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$, and $\boldsymbol{\beta}_3$ are vectors of coefficients; and u_i is an error term that may be correlated with the endogenous regressors.

We assume the existence of a set of exogenous instruments for each endogenous regressor. These sets of instruments can be the same across endogenous regressors, or they can be different. Let $\mathbf{z}_i^k$ be the vector containing the instruments for endogenous regressor y_{ik}, and let $\mathbf{z}_i = (\mathbf{z}_i^1, \mathbf{z}_i^2, \ldots, \mathbf{z}_i^p)'$ be the vector containing the instruments for all endogenous regressors in the model.

While the model is similar to those fit by instrumental-variables probit methods, the control-function approach explicitly models the endogeneity in the error term u_i. Specifically, we assume

$$\begin{aligned} P(y_{i0}=1|\mathbf{y}_i,\mathbf{x}_i,\mathbf{z}_i,\mathbf{w}_i) &= P(y_{i0}=1|\boldsymbol{\nu}_i,\mathbf{y}_i,\mathbf{x}_i,\mathbf{z}_i,\mathbf{w}_i) \\ &= \Phi(\mathbf{y}_i\boldsymbol{\beta}_1+\mathbf{x}_i\boldsymbol{\beta}_2+\mathbf{w}_i\boldsymbol{\beta}_3+\nu_i\boldsymbol{\rho}+h(\boldsymbol{\nu}_i,\mathbf{y}_i,\mathbf{x}_i,\mathbf{z}_i,\mathbf{w}_i)'\boldsymbol{\rho}_h+\epsilon_i) \end{aligned}$$

where ϵ_i is an error term unaffected by endogeneity. Here $\nu_i=(\nu_{i1},\nu_{i2},\ldots,\nu_{ip})'$ is a row vector of control functions, one for each endogenous variable, and $\boldsymbol{\rho}=(\rho_1,\rho_2,\ldots,\rho_p)$ is a vector of parameters. $h(\cdot)$ is a known vector-valued function and can include, for our purposes, interactions among the control functions in ν_i, as well as between the control functions and the exogenous or endogenous variables. $\boldsymbol{\rho}_h$ is a set of associated parameters.

For example, suppose that we have one endogenous variable y_{i1} and two instruments z_{i1}^1 and z_{i2}^1 and that $\mathbf{x}_i$ and $\mathbf{w}_i$ are empty. $h(\boldsymbol{\nu}_i,y_{i1},\mathbf{z}_i^1)$ might take the form $(\nu_{i1}z_{i1}^1,\nu_{i1}z_{i2}^1)$. We can write

$$P(y_{i0}=1|\mathbf{y}_i,\mathbf{x}_i,\mathbf{w}_i)=\Phi(y_{i1}\beta_1+\nu_{i1}\rho_1+\nu_{i1}z_{i1}^1\rho_{h1}+\nu_{i1}z_{i2}^2\rho_{h2}+\epsilon_i)$$

▷ Example 1: Single endogenous regressor, linear first stage

In practice, control functions are not observed but rather estimated. Specifically, the residuals or generalized residuals produced in first-stage regressions serve as control functions. We can model the endogenous variable y_{i1} by the linear regression

$$y_{i1}=\mathbf{x}_i\boldsymbol{\pi}_{11}+\mathbf{z}_i^1\boldsymbol{\pi}_{12}+\nu_{i1}$$

and use the estimate $\widehat{\nu}_{i1}$ as our control function for y_{i1}.

To illustrate, we revisit [ERM] **Example 3a**, where we used a fictional dataset of university students to investigate the relationship between `graduate`, an indicator for college graduation, and `hsgpa`, a variable for high school grade point average. Rather than modeling the endogenous regressor and the main outcome jointly, here we will model the endogeneity using a control function that enters the main probit model.

A variable for income (`income`) and an indicator for whether a student has roommates (`roommate`) are included as exogenous variables. An index of the competitiveness of a student's high school (`hscomp`) is included as a set of categorical instrumental variables for `hsgpa`, which is thought to be endogenous. The main probit model has the form

$$P(\texttt{graduate}_i=1)=\Phi(\beta_0+\beta_1\texttt{hsgpa}_i+\beta_2\texttt{1.roommate}_i+\beta_3\texttt{income}_i+u_i)$$

To fit this model with `cfprobit`, with a first-stage regression model of `hsgpa` on `income`, `i.hscomp`, and `i.roommate`, we could type

```
. use https://www.stata-press.com/data/r18/class10
. cfprobit graduate income i.roommate (hsgpa = i.hscomp)
```

However, in [ERM] **Example 3a**, robust standard errors are reported, and `i.roommate` appears only in the main equation. We can use the `mainonly(i.roommate)` and `vce(robust)` option to produce similar results as follows:

```
. cfprobit graduate income (hsgpa = i.hscomp), mainonly(i.roommate) vce(robust)

Iteration 0:  Log pseudolikelihood = -1670.5207
Iteration 1:  Log pseudolikelihood = -1225.1014
Iteration 2:  Log pseudolikelihood =  -1220.737
Iteration 3:  Log pseudolikelihood = -1220.7329
Iteration 4:  Log pseudolikelihood = -1220.7329

Control-function probit regression                    Number of obs =  2,500
                                                      Wald chi2(3)  = 366.50
                                                      Prob > chi2   = 0.0000

Endogenous variable model:
    Linear: hsgpa
```

graduate	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
graduate						
hsgpa	1.108869	.4212983	2.63	0.008	.2831396	1.934599
roommate						
Yes	.2835928	.0598045	4.74	0.000	.1663782	.4008074
income	.1712461	.0222423	7.70	0.000	.127652	.2148403
_cons	-3.958217	1.142227	-3.47	0.001	-6.196942	-1.719492
e.graduate						
cf(hsgpa)	1.500675	.4308452	3.48	0.000	.656234	2.345116

```
Instruments for hsgpa: 2.hscomp 3.hscomp
```

The estimates here are similar to those in [ERM] **Example 3a** but not identical, as is to be expected. The control-function procedure also gives us an estimate of the coefficient on the control function in the main equation, reported as `cf(hsgpa)`. Here `e.graduate` denotes the model for u_i, the error term in the main probit equation. This error term is modeled as a function of the control functions and, in some cases, other interaction terms involving them. In our example, a test of the hypothesis that the coefficient on `cf(hsgpa)` is different from zero can be interpreted as a test of the endogeneity of `hsgpa`.

cfprobit allows us to specify variables that interact with the control function. One interesting use of this feature, as outlined by Wooldridge (2015) in a linear control-function setting, is to specify a model with a correlated random coefficient on the endogenous variable. To do this, we include an interaction between hsgpa and the control function in the main model:

```
. cfprobit graduate income (hsgpa = i.hscomp, interact(hsgpa)),
> mainonly(i.roommate) vce(robust)
Iteration 0:  Log pseudolikelihood = -1670.5207
Iteration 1:  Log pseudolikelihood = -1227.3099
Iteration 2:  Log pseudolikelihood = -1215.8736
Iteration 3:  Log pseudolikelihood = -1215.6982
Iteration 4:  Log pseudolikelihood = -1215.6981
Control-function probit regression                    Number of obs =  2,500
                                                      Wald chi2(3)  = 385.44
                                                      Prob > chi2   = 0.0000
Endogenous variable model:
    Linear: hsgpa
```

graduate	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
graduate						
hsgpa	1.200042	.4204305	2.85	0.004	.3760133	2.02407
roommate						
Yes	.2894069	.0599019	4.83	0.000	.1720013	.4068124
income	.1729677	.0223577	7.74	0.000	.1291474	.2167881
_cons	-4.289334	1.139105	-3.77	0.000	-6.52194	-2.056729
e.graduate						
cf(hsgpa)	-2.005797	1.372688	-1.46	0.144	-4.696216	.6846215
cf(hsgpa)						
hsgpa	1.217436	.4549708	2.68	0.007	.3257092	2.109162

```
Instruments for hsgpa: 2.hscomp 3.hscomp
```

Conveniently, a test of the coefficient on the interaction term cf(hsgpa)#c.hsgpa is a valid test of whether the linear coefficient on hsgpa is random. The results suggest it is and, thus, that a random coefficient model is appropriate.

◁

Example 2: Two endogenous regressors

The fictional university student database discussed above includes an indicator variable, program, for whether a student participated in a study-skills program. We can include this as an endogenous variable in the regression, instrumenting it using indicator variables scholar, for whether the student has a scholarship, and campus, for whether the student lived on campus in their first year. We use a probit model for the first-stage regression for program.

Because there are two endogenous variables and thus two control functions, we can include the interaction of the two control functions in the main model using the cfinteract option.

```
. cfprobit graduate income (hsgpa = i.hscomp, interact(hsgpa))
> (program = i.campus i.scholar, probit interact(i.program)),
> mainonly(i.roommate) vce(robust) cfinteract

Iteration 0:  Log pseudolikelihood = -1670.5207
Iteration 1:  Log pseudolikelihood = -1097.3719
Iteration 2:  Log pseudolikelihood = -1077.9706
Iteration 3:  Log pseudolikelihood = -1077.6541
Iteration 4:  Log pseudolikelihood = -1077.6538
Iteration 5:  Log pseudolikelihood = -1077.6538

Control-function probit regression                    Number of obs =  2,500
                                                      Wald chi2(4)  = 407.05
                                                      Prob > chi2   = 0.0000

Endogenous variable models:
    Linear: hsgpa
    Probit: 1.program
```

graduate	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
graduate						
hsgpa	1.291673	.4500548	2.87	0.004	.4095819	2.173765
1.program	.4903884	.1758995	2.79	0.005	.1456318	.8351451
roommate						
Yes	.3237032	.0633824	5.11	0.000	.1994761	.4479304
income	.2166386	.0250304	8.66	0.000	.1675799	.2656973
_cons	-5.092267	1.222169	-4.17	0.000	-7.487675	-2.696859
e.graduate						
cf(hsgpa)	-2.367223	1.550839	-1.53	0.127	-5.406811	.6723648
cf(1.program)	.3197215	.1520979	2.10	0.036	.021615	.6178279
cf(1.program)						
cf(hsgpa)	.0225537	.1966956	0.11	0.909	-.3629625	.40807
cf(hsgpa)						
hsgpa	1.442845	.519374	2.78	0.005	.4248904	2.460799
cf(1.program)						
program						
0	0	(omitted)				
1	.1701306	.2206631	0.77	0.441	-.2623611	.6026222

```
Instruments for hsgpa:     2.hscomp 3.hscomp
Instruments for 1.program: 1.campus 1.scholar
```

To test the endogeneity of program, we can perform a joint test of the coefficients on cf(program) and its interactions using the postestimation command estat endogenous.

```
. estat endogenous program

Tests of endogeneity
H0: Variables are exogenous

 ( 1)  [e.graduate]cf(1.program) - [e.graduate]cf(1.program)#cf(hsgpa) = 0
 ( 2)  [e.graduate]cf(1.program) - [e.graduate]cf(1.program)#0b.program = 0
 ( 3)  [e.graduate]cf(1.program) - [e.graduate]cf(1.program)#1.program = 0
 ( 4)  [e.graduate]cf(1.program) = 0
       Constraint 4 dropped

           chi2(  3) =   12.69
         Prob > chi2 =    0.0054
```

The results suggest that program is indeed endogenous.

◁

Stored results

cfprobit stores the following in e():

Scalars

e(N)	number of observations
e(N_cds)	number of completely determined successes
e(N_cdf)	number of completely determined failures
e(k_endog)	number of endogenous variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(chi2)	χ^2
e(p)	p-value for model test
e(N_clust)	number of clusters
e(hac_lag)	HAC lag
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	cfprobit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(endog)	names of endogenous variables
e(exog)	names of exogenous variables
e(exog_main)	names of exogenous variables in main equation only
e(constant)	noconstant, if specified
e(wtype)	weight type
e(wexp)	weight expression
e(modeltypes)	model specification (linear, probit, etc.) for each endogenous regressor
e(cfinteract)	cfinteract, if specified
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(hac_kernel)	HAC kernel
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(exogr)	exogenous regressors
e(asis)	asis, if specified
e(method)	ml
e(opt)	type of optimization

e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(footnote)	program used to implement footnote display
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved
Matrices	
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

The probit model fit by cfprobit can be written as

$$P(y_{i0} = 1|\mathbf{y}_i, \mathbf{x}_i) = \Phi(\mathbf{y}_i\boldsymbol{\beta}_1 + \mathbf{x}_i\boldsymbol{\beta}_2 + u_i)$$

where y_{i0} is the dependent variable for the ith observation; $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{ip})'$ is a row vector of p endogenous regressors; $\mathbf{x}_i$ is a row vector of exogenous variables; $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are vectors of coefficients; and u_i is an error term, which may be correlated with the endogenous regressors $\mathbf{y}_i$. Here we have omitted the main equation-only variables $\mathbf{w}_i$ for convenience.

We also specify first-stage models for each endogenous regressor y_{ik} as a function of the exogenous regressors $\mathbf{x}_i$ and instruments $\mathbf{z}_i^k$. These first-stage models, and the associated control functions ν_i and their estimates $\widehat{\nu}_i$, are defined in [R] **cfregress**.

Regression estimates are produced using a modified main equation that incorporates the control-function model of the endogeneity in u_i,

$$P(y_{i0} = 1|\nu_i, \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i) = \Phi\{\mathbf{y}_i\boldsymbol{\beta}_1 + \mathbf{x}_i\boldsymbol{\beta}_2 + \nu_i\boldsymbol{\rho} + h(\widehat{\nu_i}, \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i)'\boldsymbol{\rho}_h + \epsilon_i\}$$

where $h(\cdot)$ is known, $\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_p)$, $\boldsymbol{\rho}_h$ is a vector of coefficients corresponding to the elements of $h(\widehat{\nu_i}, \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i)$, $\mathbf{z}_i = (\mathbf{z}_i^1, \mathbf{z}_i^2, \ldots, \mathbf{z}_i^p)'$, and ϵ_i is an error term. This probit model is estimated using maximum likelihood to produce estimates of the coefficients that are appropriately corrected for endogeneity.

Note that control functions enter as estimates that have been computed in a first-stage model. As a result, the standard errors returned by this procedure are incorrect. For this reason, the standard errors are computed as if the model was estimated using generalized method of moments (GMM). The GMM specification used to produce standard errors includes a set of moment conditions for the main equation, as well as a set of moment conditions for each of the first-stage models.

The error function for the dependent variable is

$$\epsilon_i(y_{i0}, \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\rho}, \boldsymbol{\rho}_h) = y_{i0}\frac{\phi(\omega)}{\Phi(\omega)} - (1 - y_{i0})\frac{\phi(-\omega)}{\Phi(-\omega)}$$

where $\omega = \mathbf{y}_i\boldsymbol{\beta}_1 + \mathbf{x}_i\boldsymbol{\beta}_2 + \nu_i(\mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i)\boldsymbol{\rho} + h(\nu_i(\mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i), \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i)'\boldsymbol{\rho}_h$.

Note that this error function is equal to the score of the probit log-likelihood function (and thus delivers the same coefficient estimates as a maximum likelihood procedure). It forms a set of moment conditions with associated instruments $\mathbf{y}_i$, $\mathbf{x}_i$, $\widehat{\boldsymbol{\nu}}_i$, and $h(\widehat{\boldsymbol{\nu}}_i, \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i)$.

Additionally, each of the control functions $\boldsymbol{\nu}_{ki}(\mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i^k)$ is taken as an error function that forms a set of moment conditions with the exogenous variables $\mathbf{x}_i$ and associated instruments $\mathbf{z}_i^k$.

Together, these moment conditions define an exactly identified model for the purpose of GMM estimation, even if there are more instruments in $\mathbf{z}_i$ than there are endogenous variables (in this sense, it is a method of moments specification). This is because each instrument in the moment conditions is associated with a unique parameter. Because the GMM model is exactly identified, the results are invariant to the choice of the GMM weight matrix.

Acknowledgment

We thank Jeffrey M. Wooldridge of the Department of Economics at Michigan State University for his extensive contributions to the literature on control-function methods.

References

Andrews, D. W. K. 1991. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica* 59: 817–858. https://doi.org/10.2307/2938229.

Gallant, A. R. 1987. *Nonlinear Statistical Models*. New York: Wiley. https://doi.org/10.1002/9780470316719.

Newey, W. K., and K. D. West. 1994. Automatic lag selection in covariance matrix estimation. *Review of Economic Studies* 61: 631–653. https://doi.org/10.2307/2297912.

Wooldridge, J. M. 2015. Control function methods in applied econometrics. *Journal of Human Resources* 50: 420–445. https://doi.org/10.3368/jhr.50.2.420.

Also see

[R] **cfprobit postestimation** — Postestimation tools for cfprobit+

[R] **cfregress** — Control-function linear regression+

[R] **ivprobit** — Probit model with continuous endogenous covariates

[R] **probit** — Probit regression

[U] 20 Estimation and postestimation commands

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on citing Stata documentation.