The HIP package

Jack Lucchetti Claudia Pigini

September 14, 2023

Abstract

The HIP package is a collection of gretl scripts to estimate probit models which may feature endogenous regressors and/or heteroskedasticity. Estimation is done via maximum likelihood under the assumption of multivariate normality.

Contents

1 Introduction **1**

2 The model **2**

3 A few examples **2**
3.1 IV probit — through a script 2
3.2 IV probit — through the GUI 3
3.3 Heteroskedastic probit 5
3.4 Let's get HIP: heteroskedasticity and endogeneity at the same time 6

4 Computational details **8**

5 Changelog **8**

A The boring stuff **9**
A.1 The loglikelihood 9
A.2 The score 9

B List of functions **10**
B.1 Model setup 10
B.2 Estimation 11
B.3 Output 11
B.4 GUI wrapper 12

C Bundle elements **13**

1 Introduction

The HIP package is a collection of gretl scripts to estimate probit models which may feature endogenous regressors and/or heteroskedasticity. Estimation is done via maximum likelihood under the assumption of multivariate normality.

Most other packages provide similar facilities separately. However, the additional computational complexity of handling, at the same time, endogeneity and the special form of conditional heteroskedasticity we deal with here is minimal, so we give a command which naturally nests the two special cases but can just as easily handle the general one.

2 The model

The model which `HIP` handles can be thought of as the union of the familiar IV-probit model and the heteroskedastic probit model, that is models that can be written in the following form:

$$y_i^* = \mathbf{Y}_i'\boldsymbol{\beta}_1 + \mathbf{X}_{1i}'\boldsymbol{\beta}_2 + \varepsilon_i = \mathbf{Z}_i'\boldsymbol{\beta} + \varepsilon_i \tag{1}$$

$$\mathbf{Y}_i = \boldsymbol{\Pi}_1'\mathbf{X}_{1i} + \boldsymbol{\Pi}_2'\mathbf{X}_{2i} + \mathbf{u}_i = \boldsymbol{\Pi}'\mathbf{X}_i + \mathbf{u}_i \tag{2}$$

$$\left(\begin{array}{c} \varepsilon_i \\ \mathbf{u}_i \end{array} \middle| \, \mathbf{X}_i, \mathbf{W}_i \right) \sim N\left[\left(\begin{array}{c} 0 \\ \mathbf{0} \end{array} \right), \left(\begin{array}{cc} \sigma_i^2 & \sigma_i\boldsymbol{\lambda}' \\ \sigma_i\boldsymbol{\lambda} & \boldsymbol{\Sigma} \end{array} \right) \right] \tag{3}$$

$$\sigma_i = \exp\{\mathbf{W}_i'\boldsymbol{\alpha}\} \tag{4}$$

The variable y_i^* is assumed to be unobservable; what is observable is $y_i = I(y_i^* > 0)$, where $I()$ is the indicator function. $\mathbf{Y}_i$ is a vector of p endogenous continuous variables and $\mathbf{X}_{1i}$ is a k_1-vector of exogenous variables; equation (2) is the reduced form for the endogenous regressors in (1), and also includes a k_2-vector of instruments $\mathbf{X}_{2i}$.

The notable feature of equation (3) (apart from the customary normality assumption) is the fact that ε_i is allowed to be conditionally heteroskedastic, with variance given by equation (4), where $\mathbf{W}_i$ is a vector of q exogenous variables. Of course, the elements of $\mathbf{W}_i$ may also be elements of $\mathbf{X}_i$. For identification purposes, though, $\mathbf{W}_i$ should not include a constant term or equivalent variables, such as for example a complete set of dummies.

Note that the familiar IV-probit model arises as a special case of the above under the constraint $\boldsymbol{\alpha} = \mathbf{0}$ whereas, in a parallel fashion, the so-called "heteroskedastic probit model" corresponds to the above model under the constraint $\boldsymbol{\lambda} = \mathbf{0}$, in which case obviously the parameters in the two equations (1) and (2) become independent and can be estimated separately.

3 A few examples

3.1 IV probit — through a script

To begin with, we'll apply IV probit to a time-honoured problem, that is female labour force participation.[1] We'll use the immortal dataset used in Mroz (1987), supplied among gretl's example datasets. We will exemplify `HIP` through a script first, and then we'll take a look at the GUI hook that `HIP` provides. Of course, in both examples we'll assume `HIP` has correctly been installed.

The script can be very simple:

```
include HIP.gfn
open mroz87.gdt --quiet

list X1 = const WE KL6
series other_inc = (FAMINC - WW*WHRS) / 1000
HIP(LFP, X1, other_inc, HE)
```

[1] Examples like the one presented here are quite common in several other software packages. Go check.

which yields:

```
Probit model with endogenous regressors
ML, using observations 1-753
Dependent Variable: LFP
Instrumented: other_inc
Instruments: const, WE, KL6, HE
Parameter covariance matrix: OPG

              coefficient   std. error     z      p-value
  ---------------------------------------------------------
  const       -1.20677      0.277614     -4.347   1.38e-05 ***
  WE           0.179911     0.0297031     6.057   1.39e-09 ***
  KL6         -0.646468     0.102047     -6.335   2.37e-10 ***
  other_inc   -0.0332341    0.0156644    -2.122   0.0339   **

Log-likelihood       -3325.8255  Akaike criterion    6671.6509
Schwarz criterion     6717.8916  Hannan-Quinn        6689.4651
Conditional ll      -465.248010  Cragg-Donald stat.     51.707

Overall test (Wald) = 73.9702 (3 df, p-value = 0.0000)
Endogeneity test (Wald) = 0.446846 (1 df, p-value = 0.5038)
```

In this case we used the function `HIP`, which takes as arguments

1. the dependent variable
2. the exogenous explanatory variables (normally as a list)
3. the endogenous explanatory variables (a list or, as in this this case, a single variable name)
4. the instruments (a list or, as in this this case, a single variable name)

The function `HIP` in fact accepts more arguments that this, but we'll leave that for later. It should also be said that the function HIP produces a gretl bundle as output, although in this example the function is called in such a way that the bundle is discarded. To store the estimated model in a bundle called "Bonham", you would call the `HIP` function like this:

```
Bonham = HIP(LFP, X1, other_inc, HE)
```

The estimate you get for standard errors uses OPG (Outer Product of Gradients) as the standard method for computing the covariance matrix of the estimates. This choice was made for the sake of performance but, as will be shown below, other methods are readily available.

The auxiliary statistics reported by `HIP` are the usual likelihood-based criteria (besides the total likelihood, the maximized value for its conditional component only is also reported—see section A.1 in the appendix for details) and the Cragg–Donald statistic as a way to check for weak instruments. The endogeneity test is a test for $\boldsymbol{\lambda} = 0$, the overall test is a test for $\boldsymbol{\beta} = 0$ (apart from the intercept).

3.2 IV probit — through the GUI

After installing `HIP` by going to *Help* > *Check for addons*, you'll find it among the other function packages installed on your box (*Tools* > *Function packages* > *On local machine*). Double-click and edit the window that appears like in Figure 1.

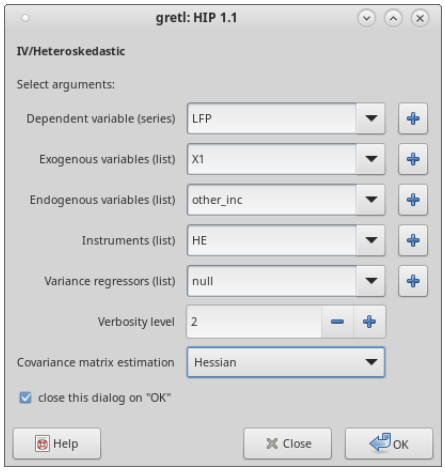


Figure 1: HIP GUI hook

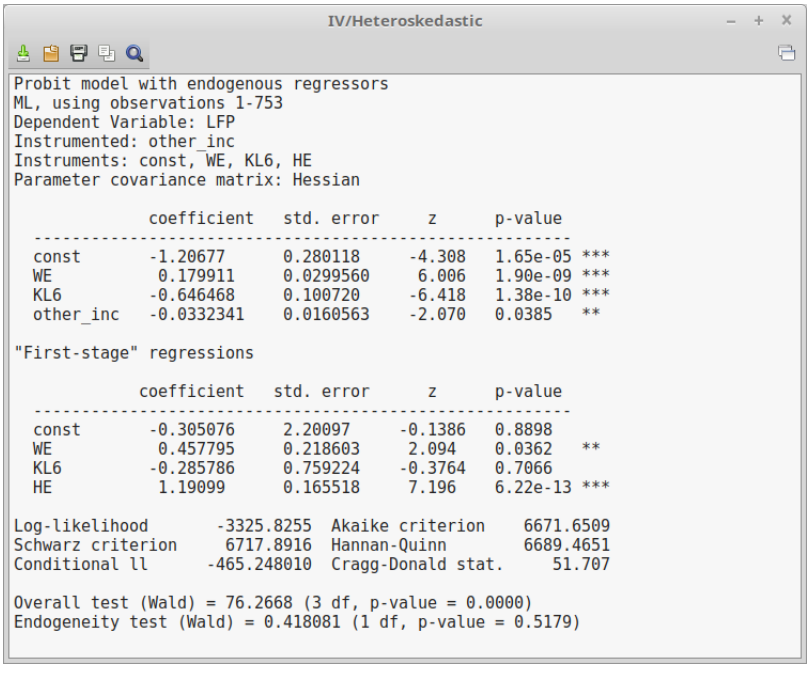


Figure 2: HIP output

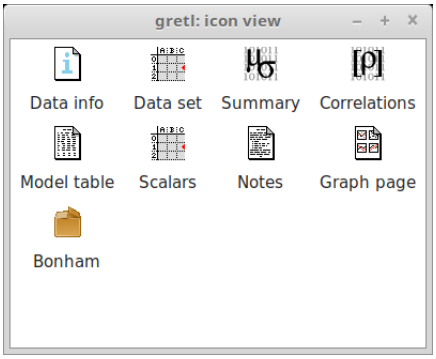


Figure 3: Icon view with a HIP bundle

Note that in this case we changed the default value of "Verbosity" from 1 to 2 and the default value of "Covariance matrix estimation" from "OPG" to "Hessian". This will have the effect of showing us the first stage equation as well, and of using the Hessian instead of the OPG as the method for computing standard errors. All this is apparent in Figure 2.

By using the *Save* menu, you can choose the individual elements of the bundle to store away for later use if you want. Alternatively, you can save the bundle as a model via the *File* > *Save to session as icon* menu entry. If you do, assuming that you called your bundle "Bonham" again, then it will show in the "Icon view" gretl window, together with other session elements you want to keep (see Figure 3).

3.3 Heteroskedastic probit

Here, we'll replicate the example given in William Greene's textbook (7th edition), which also uses Mroz's dataset. The script goes like this:

```
include HIP.gfn

open mroz87.gdt --quiet
series WA2 = WA^2
series KIDS = (KL6 + K618)>0
income = FAMINC /10000

list X = const WA WA2 income KIDS WE
list Z = income KIDS

Mitchell = HIP_setup(LFP, X, null, null, Z)
HIP_setoption(&Mitchell, "vcvmeth", 1)

set stopwatch
HIP_estimate(&Mitchell)
printf "Elapsed time = %g seconds\n", $stopwatch

HIP_printout(&Mitchell)
```

Note that in this case we did not use the `HIP` function, but instead we split its workload between four separate functions:

HIP_setup Sets up the model: basically, it has the same parameters as the all-rounder `HIP` function seen above. Returns a bundle.

HIP_setoption Set some details of the estimation procedure; in this case, we used it to compute standard errors using the inverse Hessian instead of the OPG matrix, so as to match exactly the figures reported in Greene's book.

HIP_estimate Estimates the model: takes as argument the bundle address, plus an optional scalar for the verbosity.

HIP_printout Prints out the results contained in the bundle.

This division of tasks may be convenient at times, because it gives you finer control over "what happens if". For example, the Cragg–Donald statistic gets computed during the initialization of the bundle and you may wish to decide whether to proceed with estimation or not depending on how strong your instruments are.

The output, replicating table 17.7 in Greene's textbook, should look like this:

```
Heteroskedastic probit model
ML, using observations 1-753
Dependent Variable: LFP
Parameter covariance matrix: Hessian

            coefficient   std. error     z      p-value
  ---------------------------------------------------------
  const      -6.02985      2.49810      -2.414   0.0158  **
  WA          0.264291     0.118159      2.237   0.0253  **
  WA2        -0.00362838   0.00143387   -2.530   0.0114  **
  income      0.424441     0.221839      1.913   0.0557  *
  KIDS       -0.879093     0.302753     -2.904   0.0037  ***
  WE          0.140149     0.0518536     2.703   0.0069  ***

Variance

            coefficient   std. error      z      p-value
  ---------------------------------------------------------
  KIDS        -0.140752     0.323745    -0.4348   0.6637
  income       0.312918     0.122810     2.548    0.0108  **

Log-likelihood        -487.6356  Akaike criterion     991.2712
Schwarz criterion     1028.2637  Hannan-Quinn        1005.5225

Overall test (Wald) = 14.5557 (5 df, p-value = 0.0124)
Heteroskedasticity test (LR) = 6.42453 (2 df, p-value = 0.0403)
Chesher and Irish normality test = 6.23055 (2 df, p-value = 0.0444)
```

3.4 Let's get HIP: heteroskedasticity and endogeneity at the same time

The script goes:

```
set verbose off
include HIP.gfn

open mroz87.gdt -q
```

```
list EXOG =  const WA CIT K618
list ENDOG = WE
list ADDIN =  WMED WFED
list HETVAR = HW

Paice = HIP(LFP, EXOG, ENDOG, ADDIN, HETVAR, 2)
```

In this case, the "2" tells HIP to be moderately verbose: don't print out all the iterations, but show us the "first stage" coefficients. The output is as follows:

```
Heteroskedastic probit model with endogenous regressors
ML, using observations 1-753
Dependent Variable: LFP
Instrumented: WE
Instruments: const, WA, CIT, K618, WMED, WFED
Parameter covariance matrix: OPG

            coefficient   std. error      z      p-value
  ---------------------------------------------------------
  const      -0.551804     1.36344      -0.4047   0.6857
  WA         -0.0304390    0.0172559    -1.764    0.0777  *
  CIT        -0.0242784    0.208991     -0.1162   0.9075
  K618       -0.0927252    0.0896549    -1.034    0.3010
  WE          0.199646     0.101330      1.970    0.0488  **

Variance

            coefficient   std. error     z     p-value
  ------------------------------------------------------
  HW          0.117934     0.0571806    2.062   0.0392  **

"First-stage" regressions

            coefficient   std. error     z      p-value
  ---------------------------------------------------------
  const       9.68554      0.586171     16.52    2.49e-61 ***
  WA         -0.0159435    0.0104384    -1.527   0.1267
  CIT         0.495907     0.152627      3.249   0.0012   ***
  K618       -0.136765     0.0612498    -2.233   0.0256   **
  WMED        0.180089     0.0265972     6.771   1.28e-11 ***
  WFED        0.168085     0.0253072     6.642   3.10e-11 ***

Log-likelihood       -2069.9119  Akaike criterion    4167.8239
Schwarz criterion     4232.5608  Hannan-Quinn        4192.7637
Conditional ll      -494.848818  Cragg-Donald stat.    103.337

Overall test (Wald) = 6.36207 (4 df, p-value = 0.1737)
Endogeneity test (Wald) = 0.509859 (1 df, p-value = 0.4752)
Test for overidentifying restrictions (LM) = 9.15786 (1 df, p-value = 0.0025)
Heteroskedasticity test (Wald) = 4.25379 (1 df, p-value = 0.0392)
```

4 Computational details

`HIP` uses the analytical score and BFGS as the preferred optimization method. The analytical Hessian is not implemented yet, but may be in the future.

Like other estimators that depend on numerical methods, `HIP` can sometimes run into numerical problems, leading to non-convergence. If this happens, here are some points to consider.

- Checking exactly what happens during maximization can be very informative; try setting the verbosity parameter to 3.
- Scaling of the data (especially $\mathbf{Y}_i$) can be an issue; we do our best, but hey, give us a hand (for example, multiply or divide by 1000 depending of the original scale of the data).
- Weak instruments: in some cases, there's little that can be done; see for example the artificially-generated dataset contained in the `MonteCarlo.inp` example script, contained in the examples directory. We do some heuristics, but we're not omnipotent.

5 Changelog

1.1 Guard against the inclusion of a constant in the `HETVAR` list. Fix a typo in an error message. Modernise internal syntax in a few places.

1.0 Initial release

References

Chesher, A. and M. Irish (1987) 'Residual analysis in the grouped and censored normal linear model', *Journal of Econometrics* 34: 33–61.

Cragg, J. G. and S. G. Donald (1993) 'Testing identifiability and specification in instrumental variable models', *Econometric Theory* 9: 222–240.

Mroz, T. (1987) 'The sensitivity of an empirical model of married women's hours of work to economic and statistical assumptions', *Econometrica* 5: 765–799. URL https://doi.org/10.2307/1911029.

Rivers, D. and Q. H. Vuong (1988) 'Limited information estimators and exogeneity tests for simultaneous probit models', *Journal of Econometrics* 39(3): 347–366.

A The boring stuff

For computational purposes, we reparametrize the model using the Cholesky decomposition $\mathbf{\Sigma}^{-1} = CC'$. Moreover, by defining the quantities below it is possible to reparametrize the joint density in a computationally convenient way:

$$\begin{aligned}
\nu_i &= \frac{1}{\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}}\left(\frac{\mathbf{Z}_i'\boldsymbol{\beta}}{\sigma_i} + \boldsymbol{\omega}_i'\boldsymbol{\psi}\right) \\
\boldsymbol{\psi} &= C'\boldsymbol{\lambda} \\
\boldsymbol{\omega}_i &= C'\left(\mathbf{Y}_i - \mathbf{\Pi}\mathbf{X}_i\right) \\
\boldsymbol{\pi} &= \text{vec}\left(\mathbf{\Pi}\right) \\
\mathbf{c} &= \text{vech}(C)
\end{aligned}$$

The estimable parameters are $\theta' = [\boldsymbol{\beta}', \boldsymbol{\alpha}', \boldsymbol{\pi}', \boldsymbol{\psi}', \mathbf{c}']$

A.1 The loglikelihood

As usual in such models, we divide the loglikelihood for each observation into a marginal and a conditional component:

$$\begin{aligned}
\ell_i &= \ell_i^m + \ell_i^c \\
\ell_i^c &= \ln P(y_i|\mathbf{X}_i, \mathbf{W}_i, \mathbf{u}_i) \\
\ell_i^m &= \ln f(\mathbf{u}_i|\mathbf{X}_i, \mathbf{W}_i)
\end{aligned}$$

The marginal component is nothing but an ordinary Gaussian loglikelihood:

$$\ell_i^m = -\frac{p}{2}\ln(2\pi) + \sum_{j=1}^{p}\ln c_{jj} - \frac{1}{2}\boldsymbol{\omega}_i'\boldsymbol{\omega}_i$$

The conditional component is itself rather simple:

$$\ell_i^c = y_i \ln\Phi(\nu_i) + (1-y_i)\ln\left[1-\Phi\left(\nu_i\right)\right] \tag{5}$$

The only feature that sets ℓ_i^c apart from an ordinary probit loglikelihood is that the index function depends non-linearly on some of the parameters of the model, unless $\boldsymbol{\alpha}$ and $\boldsymbol{\psi}$ are both zero.

A.2 The score

The analytical score will be derived in steps: first the marginal component, then the conditional component. Of course, the chain rule will be very useful.

Note first that the marginal component only depends on $\boldsymbol{\pi}$ (through $\boldsymbol{\omega}_i$) and $\mathbf{c}$. Hence,

$$\frac{\partial\ell_i^m}{\partial\boldsymbol{\pi}} = \frac{\partial\ell_i^m}{\partial\boldsymbol{\omega}_i}\frac{\partial\boldsymbol{\omega}_i}{\partial\boldsymbol{\pi}} = \boldsymbol{\omega}_i'\left(\mathbf{X}_i'\otimes C'\right) = \mathbf{X}_i'\otimes\left(C\boldsymbol{\omega}_i\right)'$$

and

$$\frac{\partial\ell_i^m}{\partial\mathbf{c}} = \tilde{\mathbf{c}}' - \boldsymbol{\omega}_i'\frac{\partial\boldsymbol{\omega}_i}{\partial\mathbf{c}} = \tilde{\mathbf{c}}' - \left[\boldsymbol{\omega}_i'\otimes\left(\mathbf{Y}_i - \mathbf{\Pi}\mathbf{X}_i\right)'\right]S$$

where $\tilde{\mathbf{c}}$ is defined as $\text{vech}\left[\left(I\odot C\right)^{-1}\right]$ and S is a selection matrix $S = \frac{\partial\text{vec}(C)}{\partial\text{vech}(C)}$.

For the purpose of computing the score for the conditional component, note that ℓ_i^c depends on the parameters only through the index function ν_i, so $\frac{\partial \ell_i^c}{\partial \theta}$ can be evaluated as

$$\frac{\partial \ell_i^c}{\partial \theta} = \frac{\partial \ell_i^c}{\partial \nu_i}\frac{\partial \nu_i}{\partial \theta};$$

define $\mu(\nu_i)$ as

$$\mu(\nu_i) = \frac{\partial \ell_i^c}{\partial \nu_i} = y_i \frac{\phi(\nu_i)}{\Phi(\nu_i)} - (1-y_i)\frac{\phi(\nu_i)}{1-\Phi(\nu_i)}$$

which is the customary (signed) inverse Mills ratio. Then,

$$\begin{aligned}
\frac{\partial \nu_i}{\partial \boldsymbol{\beta}} &= \frac{1}{\sigma_i\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}}\mathbf{Z}_i' \\
\frac{\partial \nu_i}{\partial \boldsymbol{\alpha}} &= \frac{\partial \nu_i}{\partial \sigma_i}\frac{\partial \sigma_i}{\partial \boldsymbol{\alpha}} = \left[-\frac{\mathbf{Z}_i'\boldsymbol{\beta}}{\sigma_i^2\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}}\right]\sigma_i\mathbf{W}_i' = -\left(\frac{\mathbf{Z}_i'\boldsymbol{\beta}}{\sigma_i\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}}\right)\mathbf{W}_i' \\
\frac{\partial \nu_i}{\partial \boldsymbol{\psi}} &= \frac{1}{\sigma_i^2(1-\boldsymbol{\psi}'\boldsymbol{\psi})}\left[\sigma_i^2\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}\boldsymbol{\omega}_i' - \frac{\sigma_i^2}{2}\nu_i(-2\cdot\boldsymbol{\psi}')\right] = \frac{\boldsymbol{\omega}_i'}{\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}} + \frac{\nu_i\boldsymbol{\psi}'}{1-\boldsymbol{\psi}'\boldsymbol{\psi}} \\
\frac{\partial \nu_i}{\partial \mathbf{c}} &= \frac{\boldsymbol{\psi}'}{\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}}\frac{\partial \boldsymbol{\omega}_i}{\partial \mathbf{c}} \\
\frac{\partial \nu_i}{\partial \boldsymbol{\pi}} &= \frac{\boldsymbol{\psi}'}{\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}}\frac{\partial \boldsymbol{\omega}_i}{\partial \boldsymbol{\pi}} = -\frac{\boldsymbol{\psi}'}{\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}}(\mathbf{X}_i' \otimes C') = \frac{1}{\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}}[\mathbf{X}_i' \otimes (C\boldsymbol{\psi})']
\end{aligned}$$

As a consequence, the score with respect to $\mathbf{c}$ and $\boldsymbol{\pi}$ may be written as

$$\begin{aligned}
\frac{\partial \ell_i}{\partial \mathbf{c}} = \frac{\partial \ell_i^m}{\partial \mathbf{c}} + \frac{\partial \ell_i^c}{\partial \mathbf{c}} &= \tilde{\mathbf{c}}' + \left(\frac{\boldsymbol{\psi}'}{\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}} - \boldsymbol{\omega}_i'\right)\frac{\partial \boldsymbol{\omega}_i}{\partial \mathbf{c}} = \\
&= \tilde{\mathbf{c}}' + \left(\frac{\boldsymbol{\psi}'}{\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}} - \boldsymbol{\omega}_i'\right)[I \otimes (\mathbf{Y}_i - \boldsymbol{\Pi}\mathbf{X}_i)'] = \\
&= \tilde{\mathbf{c}}' + \left[\left(\frac{\boldsymbol{\psi}'}{\sqrt{1-\boldsymbol{\psi}'\boldsymbol{\psi}}} - \boldsymbol{\omega}_i'\right) \otimes (\mathbf{Y}_i - \boldsymbol{\Pi}\mathbf{X}_i)'\right]
\end{aligned}$$

B List of functions

B.1 Model setup

`HIP_setoption(bundle *b, string opt, scalar value)`

Return type : `scalar`

`b` : pointer to a bundle containing the model to be estimated, as created by `HIP_setup`;

`opt` : string, the option to set;

`value` : scalar, the option value

This function sets up an option for estimation of the model, so it is typically used after `HIP_setup` and before `HIP_estimate`. At present, the possible values for the `opt` field are "verbose" (possible values: 0 to 3) and "vcvmeth" (possible values: 0 to 2).

For "verbose", the meaning is: 0 = operate silently, 1 = standard output (default choice), 2 = print the first stage too for IV estimation and 3 = print out ML iterations. For "vcvmethod", the meaning is: 0 = OPG (default), 1 = Hessian, 2 = Sandwich-robust.

```
HIP_setup(series y, list EXOG, list ENDOG[null], list ADDIN[null],
            list HETVAR[null])
```

Return type : `bundle`

`y` : a series containing y_i, the dependent binary variable; **(required)**

`EXOG` : a list containing the exogenous variables $\mathbf{X}_{1i}$ in $\mathbf{X}_i$ in equation (1); **(required)**

`ENDOG` : a list containing the exogenous variables $\mathbf{Y}_i$ in equations (1)–(2)

`ADDIN` a list containing the additional instruments $\mathbf{X}_{2i}$ in $\mathbf{X}_i$ in equation (2)

`HETVAR` a list containing the variables $\mathbf{W}_i$ of the skedastic function in equation (1)

This function sets the model up so that it can be subsequently estimated via `HIP_estimate`.

B.2 Estimation

```
HIP_estimate(bundle *b)
```

Return type : `scalar`

`b` : a model bundle in pointer form, as created by `HIP_setup`.

General estimation function. It fills the bundle with the estimated coefficients and many other quantities of interest.

B.3 Output

```
HIP_printout(bundle *b
```

Return type : none.

`b` : a model bundle in pointer form, as created by `HIP_setup` and filled up by `HIP_estimate`.

Prints out a model. Note: this function assumes that the bundle it refers to contains a model that has already been estimated. No checks are performed.

B.4 GUI wrapper

```
function bundle HIP(series y, list EXOG, list ENDOG[null],
      list ADDIN[null], list HETVAR[null],
      int v[0:3:1], int s[0:2:0])
```

Using the same argument descriptions as `HIP_setup`, after checking the rank condition (if estimating instrumental variables probit), it calls:

1. `HIP_setup`
2. `HIP_setoption`
3. `HIP_estimate`
4. `HIP_printout`

The parameter `v` controls the verbosity level: 0 = quiet, 1 = main equation only, 2 = first stages, 3 = mle verbose.

C Bundle elements

Name	Type	Purpose
		Model descriptors
n	scalar	number of observations
het	scalar	acting as a Boolean switch, Heteroskedastic probit
iv	scalar	acting as a Boolean switch, Instrumental Variables probit
T	scalar	number of observations used
t1	scalar	first observation used
t2	scalar	last observation used
		Data
depvar	series	dependent variable
mEXOG	matrix	exogenous regressors
mk1	matrix	number of exogenous regressors
mENDOG	matrix	endogenous regressors
mp	matrix	number of endogenous regressors
mADDIN	matrix	additional instruments
mk2	matrix	number of additional instruments
mHETVAR	matrix	variance regressors
mq	matrix	number of variance regressors
mZ	matrix	total regressors
mh	matrix	number of total regressors
mX	matrix	total instruments
mk	matrix	number of total instruments
		Strings
depvarname	string	dependent variable name
mEXOGnames	string	exogenous regressors names
mENDOGnames	string	endogenous regressors names
mADDINnames	string	additional instruments names
mHETVARnames	string	variance regressors names
mZnames	string	total regressors names
mXnames	string	total instruments names
		Estimation parameters
vcvtype	scalar	acting as an integer, method for estimating the covariance matrix: 0 = OPG (default), 1 = empirical Hessian, 2 = Sandwich

	Estimation results	
`errcode`	scalar	error code from `catch`
`uhat`	series	first stage residuals (Rivers and Vuong, 1988)
`rescale`	matrix	square root of the diagonal elements of first stage residuals covariance matrix
`lnl0`	scalar	second stage log-likelihood (Rivers and Vuong, 1988)
`theta`	matrix	coefficients
`VCVtheta`	matrix	covariance matrix
`lnl1`	scalar	log-likelihood
`lnl1m`	scalar	marginal log-likelihood (if `iv`)
`lnl1c`	scalar	conditional log-likelihood (if `iv`)
`llt`	series	log-likelihood
`SCORE`	matrix	score matrix by observation
`infocrit`	matrix	information criteria
	Diagnostics[2]	
`WaldAll`	matrix	Wald overall test
`WaldEnd`	matrix	Wald endogeneity test
`LMOverid`	matrix	LM test for overidentifying restrictions
`HETtest`	matrix	if `iv` Wald test, else LR test of Heterosckedasticity
`CraggDondald`	scalar	Cragg and Donald (1993) statistic for weak instruments
`normtest`	matrix	Conditional moment test for normality of ε_i Chesher and Irish (1987)