The Stata Journal (2019)
19, Number 1, pp. 87–111 DOI: 10.1177/1536867X19830893

Estimation methods in the presence of corner solutions

Alfonso Sánchez-Peñalver
University of South Florida
Tampa, FL
alfonso.statalist@gmail.com

Abstract. In this article, I introduce a new command, `nehurdle`, that collects maximum likelihood estimators for linear, exponential, homoskedastic, and heteroskedastic tobit; truncated hurdle; and type II tobit models that involve explained variables with corner solutions. I review what a corner solution is as well as the assumptions of the mentioned models.

Keywords: st0550, nehurdle, two-part, truncated hurdle, tobit, heckman, type II tobit, corner solutions, hurdle estimation

1 Introduction

In economics, a corner solution is the situation that happens when a person chooses to not do some activity in favor of doing some other activity, for example, to not give money to charity in favor of spending money on food. There may be many different reasons why this happens. The person may totally dislike the activity and would never want to do it unless compensated for it; in economics, this activity would be called a "bad". Even if the activity is a "good" in economics (that is, even if the person likes the activity), the person may still choose to not do it. Giving to charity is mostly appreciated by everybody, yet many people choose to not do it simply because, given their respective circumstances, they prefer to do other activities more. We, therefore, will observe a value of 0 in the variable up to a certain value of the other variable, at which point the person will decide to spend money on the activity.

Figure 1 illustrates this situation and is produced using the following code:

```
. set obs 1
number of observations (_N) was 0, now 1
. generate x1 = 4
. generate y1 = 0
. generate x2 = 5.29
. generate y2 = 0
. generate x3 = 6.25
. generate y3 = 1.5
```



```
. twoway (function y = 4 - x, range(0 4) lpattern(dash))
>        (function y = 7.75 - x, range(0 7.75) lpattern(dash))
>        (function y = 5.29 - x, range(0 5.29) lpattern(dash))
>        (function y = 10 - 5*sqrt(x), range(0 4) lpattern(solid))
>        (function y = 11.5 - 5*sqrt(x), range(0.09 5.29) lpattern(solid))
>        (function y = 14 - 5*sqrt(x), range(0.64 7.84) lpattern(solid))
>        (scatter y1 x1, msymbol(O) mcolor(black))
>        (scatter y2 x2, msymbol(O) mcolor(black))
>        (scatter y3 x3, msymbol(O) mcolor(black)), legend(off)
>        plotregion(margin(zero)) xscale(range(0 8)) yscale(range(0 10))
>        xlabel(0(2)8) ylabel(0(2)10) xmtick(0(1)8) ymtick(0(1)10)
>        xtitle("Food ($ Thousands)") ytitle("Donations ($ Thousands)")
```

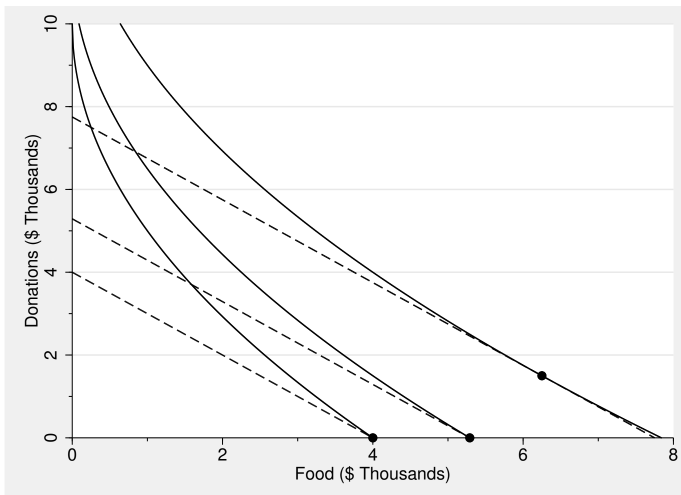

Figure 1. Solutions under different incomes

Consider that Doris is deciding how much money to give to charity and how much to spend on food, both “goods”. The dashed lines collect the maximum different combinations of the two goods, donations and food, that Doris can attain at three different levels of income. The solid curves are the combinations of the two goods that she likes equally at these three levels of income. The solid dots indicate where the dashed lines and solid curves meet. We call the dashed lines her budget lines and the solid curves her indifference curves. Because both food and giving are goods, Doris prefers to do more of both to less, so she always reaches the indifference curve that touches her budget line at the furthest point from the origin. You can see that there is a range of income values for which Doris chooses to not give to charity, but when income reaches a certain value, she starts giving. There is, then, a hurdle that she must overcome to start giving.

In econometrics, a variable with corner solutions is a particular case of a censored variable. The censoring is not caused by the way the data are collected, as is the case for other censored variables, but rather by the nature of the choice faced by the giver. The observed variable (donations in our example) will show a value of 0 when the latent variable—that is, the one that would reflect the actual amount the giver is willing to give—is below a certain value. This certain value may or may not be 0, depending on the good, its price, and our preferences. The fact remains that the observed 0 may represent an actual 0 as well as a range of values that the uncensored variable would take.

In this article, I introduce a new command, `nehurdle`, that fits models involving explained variables with corner solutions.[1] `nehurdle` collects maximum likelihood estimators for tobit, truncated hurdle, and type II tobit models.[2] Stata commands `tobit`, `churdle`, and `heckman` can be used to estimate some specifications of the three mentioned models, respectively. `nehurdle` adds the possibility to model heteroskedasticity and an exponential specification to `tobit` and `heckman` for the case of corner solutions. This functionality is already available in `churdle`, so for the truncated hurdle model, `nehurdle` offers another option as well as different functionality for the postestimation `predict` and `margins` commands. However, `churdle` is only available in Stata 14 or later, while `nehurdle` works with Stata 11 or later; so if you have a version of Stata prior to 14, `nehurdle` provides a unique option for a truncated hurdle estimator as well.

The article is organized as follows: section 2 presents `nehurdle`'s syntax, section 3 explains the differences between the three models and illustrates how to use `nehurdle` for each case, as well as how to get estimates of the partial effects after estimation, and section 4 concludes.

2 The nehurdle command

2.1 Syntax

`nehurdle` is implemented as an `lf2 ml` estimator (see [R] **ml**), so it has programmed analytical gradients and Hessians for all the estimators. Its general syntax is

`nehurdle` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, {`tobit`|`trunc`|`heckman`}
`coeflegend` `exponential` `exposure(`*varname*`)`
`het(`*indepvars* [, `noconstant`]`)` `level(`*#*`)` `noconstant` `nolog`
`offset(`*varname*`)` `vce(`*vcetype*`)` *ml_options* `select(`[*indepvars*]
[, `het(`*indepvars*`)` `noconstant` `exposure(`*varname*`)` `offset(`*varname*`)`]`)`]

We use the command name followed by the name of the explained (dependent) variable. Everything else is optional; if nothing else is specified, `nehurdle` will estimate a linear, homoskedastic, truncated hurdle model with a constant in the selection equation and a constant in the value equation. After the explained variable, we list the explanatory (independent) variables in the value equation. The command allows the use of `if` or `in` to specify which observations to use, and it also allows for the use of `fweight`, `iweight`, and `pweight`.

1. I chose the name `nehurdle` to reflect that the estimators collected in the command all fit a different type of hurdle model, so the name can be interpreted as saying "n-e" (any) hurdle.
2. See chapter 17 in Wooldridge (2010) for a deeper understanding of these models.

2.2 Options

`tobit` tells `nehurdle` to use the tobit estimator. `tobit`, `trunc`, and `heckman` are mutually exclusive.

`trunc` tells `nehurdle` to use the truncated hurdle estimator. This is `nehurdle`'s default estimator. `tobit`, `trunc`, and `heckman` are mutually exclusive.

`heckman` tells `nehurdle` to use the type II tobit estimator. `tobit`, `trunc`, and `heckman` are mutually exclusive.

`coeflegend` displays a legend instead of statistics.

`exponential` indicates an exponential value equation.

`exposure(`*varname*`)` includes ln(*varname*) in the model for the value equation with its coefficient constrained to 1.

`het(`*indepvars* [`, noconstant`]`)` allows specification of the explanatory variables as well as whether to suppress the constant for the heteroskedasticity of the value equation.

`level(`*#*`)` specifies the confidence level for the confidence intervals. The default is `level(95)`.

`noconstant` suppresses the constant in the value equation.

`nolog` suppresses maximum likelihood's iteration log.

`offset(`*varname*`)` adds *varname* to the model of the value equation with its coefficient constrained to 1.

`vce(`*vcetype*`)` specifies the estimator for the variance–covariance matrix, where *vcetype* can be `cluster` *clustvar*, `oim`, `opg`, or `robust`.

ml_options: Also common to the three estimators are the following `ml` and `maximize` options: `collinear`, `constraints(`*numlist* | *matname*`)`, `difficult`, `gradient`, `hessian`, `iterate(`*#*`)`, `ltolerance(`*#*`)`, `nocnsnotes`, `nonrtolerance`, `nrtolerance(`*#*`)`, `qtolerance(`*#*`)`, `showtolerance`, `showstep`, `technique(`*algorithm_spec*`)`, `tolerance(`*#*`)`, `trace`. See [R] **maximize** and [R] **ml** for explanations on these options.

The truncated hurdle and type II tobit estimators further have the `select()` option to specify the selection equation. This option has the following syntax:

`select(`[*indepvars*] [`, het(`*indepvars*`) noconstant exposure(`*varname*`)` `offset(`*varname*`)`]`)`

This option is not required when using either of the estimators for which it is available. By default, `nehurdle` assumes that the explanatory variables for the selection equation are the same as for the value equation, that the selection equation is to be estimated with a constant term, that its errors are homoskedastic, and that no variable is to be exposed or offset. If you do specify the `select()` option, specifying the explanatory variables is still optional; if you do not specify them, then those of the value equation will be used. The `het(`*indepvars*`)` option allows you to specify the explanatory variables for the heteroskedasticity of the selection equation. The rest of the options have the same functionality for the selection equation as their counterparts explained above do for the value equation.

3 The models

In this section, we consider each estimator and the assumptions each makes about the generating process of the explained variable. To better illustrate these assumptions, we generate a particular dataset for each of the estimators that assumes homoskedastic errors in all equations and a linear specification of the value equation. We will also consider how to use the exponential and heteroskedasticity specifications `nehurdle` allows for. For those estimations, we use a subsample of single households from the 2005 Family Data of the Panel Study of Income Dynamics (PSID) survey, conducted by the Institute for Social Research at the University of Michigan. The purpose of this dataset is to illustrate the use of the command, so here I simply present the description of the different variables used as well as some descriptive statistics:

```
. use psid2005

. describe

Contains data from psid2005.dta
  obs:         2,645
 vars:            10                          28 Jul 2017 16:51
 size:       105,800
--------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
--------------------------------------------------------------------------------
education       float   %9.0g                 Education in Years
donation        float   %9.0g                 Money Donation
spi             float   %9.0g                 State Price Index
rtincome        float   %9.0g                 Real Total Income in Thousands
rpdon           float   %9.0g                 Real Price of Money Donations
rexogdon        float   %9.0g                 Real Exogenous Donations in
                                                Thousands
rexogtax        float   %9.0g                 Real Exogenous Taxes in Thousands
age             float   %13.0g     age        Age of Giver in Years
npeople         float   %9.0g                 Number of People in the Household
female          float   %9.0g                 1 = Female
--------------------------------------------------------------------------------
Sorted by:
```

```
. summarize
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
   education |      2,645    12.66011    2.456385          0         17
    donation |      2,645    538.2038     1585.03          0      46450
         spi |      2,645    .9635085    .1554936      .6641     1.3096
   rtincome |      2,645      23.747    28.43223  -4.718094   390.3033
       rpdon |      2,645    1.030528    .1790905   .5154245   1.505797
-------------+---------------------------------------------------------
    rexogdon |      2,645    288.1902    124.1845   .4519108   507.9555
    rexogtax |      2,645    2771.649    1522.377   11.41752   6711.577
         age |      2,645    1.260113    1.006981          0          3
     npeople |      2,645    2.370132    1.586707          1         13
      female |      2,645    .6627599     .472857          0          1
```

3.1 Tobit

The first model we consider is the tobit model.[3] Consider that the optimal amount of donations a person i gives, D_i^*, has the following relationship with income:

$$D_i^* = \beta_0 + \beta_1 \text{income}_i + u_i \tag{1}$$

where D_i^* represents the optimal amount of donations by individual i and $u_i \sim N(0, \sigma_u^2)$ represents unobserved characteristics that are not correlated with income. The tobit model assumes that the selection process that determines whether we give or not is the same process that determines the value we give and assumes that the actual censoring value is 0. This means that instead of observing the full spectrum of D_i^*, we observe

$$D_i = \begin{cases} 0 & \text{if } D_i^* \leq 0 \\ D_i^* & \text{otherwise} \end{cases}$$

D_i^* is the uncensored, latent variable, and D_i is the censored, observed variable. The following code generates data based on these assumptions:

```
. set obs 10000
number of observations (_N) was 0, now 10,000
. set seed 14051969
. // Income lognormally distributed in thousands
. generate double income = exp(rnormal(10.31,0.60))/1000
. label variable income "Income ($ Thousands)"
. generate double dstar = -5 + 0.25 * income + rnormal(0,2)
. generate double donations = dstar
. replace donations = 0 if dstar < 0
(2,644 real changes made)
```

Income is lognormally distributed and expressed in thousands per annum. `dstar`, the uncensored variable that is expressed in dollars per annum, increases with income at a

3. This model was proposed by Tobin (1958).

constant rate of 25 cents per $\$1,000$ of income, has an intercept of $-\$5$, and is normally distributed with a standard deviation of $2.

The following ordinary least-squares (OLS) estimation allows us to see how OLS estimators are biased and inconsistent in this case.

```
. regress donations income, noheader level(99)

   donations |      Coef.   Std. Err.      t    P>|t|     [99% Conf. Interval]
-------------+----------------------------------------------------------------
      income |   .2304058   .0007144   322.53   0.000     .2285654    .2322462
       _cons |  -3.788923   .0312764  -121.14   0.000    -3.869501   -3.708345
```

The true parameter values, -5 for the intercept and 0.25 for the slope, are not included in the confidence interval for the respective parameter, which shows that the estimators are biased and inconsistent for the uncensored mean. The results also illustrate the direction of the OLS bias. With the negative values censored at 0, the estimator of the intercept is biased upward because the censoring increases the overall average. For the censored observations, the true relationship between donations and income is broken, that is, the slope for those observations is actually 0. That is why the slope estimator is biased toward 0, which in this case means that it is biased downward.

The following code shows how to use `nehurdle` to do the tobit estimation.[4]

```
. nehurdle donations income, tobit nolog
Tobit                                           Number of Obs.     =     10000
                                                Censored Obs.      =      2644
                                                Uncensored Obs.    =      7356
                                                Wald chi2(1)       =  81695.87
                                                Prob > chi2        =    0.0000
Log Likelihood        =   -16992.262            Pseudo R-squared   =    0.9235

   donations |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      income |    .249088   .0008715   285.82   0.000     .2473799     .250796
       _cons |  -5.000697   .0409269  -122.19   0.000    -5.080912   -4.920482
-------------+----------------------------------------------------------------
    /lnsigma |   .6893328   .0082895    83.16   0.000     .6730857      .70558
-------------+----------------------------------------------------------------
       sigma |   1.992386   .0165159                      1.960277    2.025021
```

We see that the estimates are very close to the true parameters we used to generate `dstar`. We are, thus, observing the estimates of the coefficients in the uncensored variable's mean. What does the estimate of the slope on income tell us? In reality, it is telling us that an increase in $1,000 in income will increase the uncensored variable's mean by 25 cents; so for those who are currently giving money, an increase of $1,000 in income will increase their donation by an average of 25 cents.

4. The following `tobit` command produces the same results: `tobit donations income, ll`.

When dealing with variables with corner solutions, we are really interested in the censored variable because it is the one we observe in real life. We, therefore, need the partial effects on the censored mean, not the uncensored one. In Stata, we use `margins` with the `dydx()` option to get estimates of partial effects. We have two possible estimators for the representative partial effects on a random person: the average partial effect (APE) and the partial effect at the means (PEM). The APE computes the arithmetic mean of the partial effects on each observation of the sample, and this is the default that `margins` uses when we do not specify any `at()` options. The PEM computes the partial effect using the arithmetic means of the appropriate explanatory variables, and it can be obtained by using the `atmeans` option. In this article, I show the APE because the censored mean is a nonlinear function of the explanatory variables.

```
. margins, dydx(*) predict(ycen)
Average marginal effects                        Number of obs     =     10,000
Model VCE    : OIM

Expression   : Prediction of censored mean E(donations|x), predict(ycen)
dy/dx w.r.t. : income

------------------------------------------------------------------------------
             |            Delta-method
             |      dy/dx   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      income |   .1834681   .0005569   329.47   0.000     .1823767    .1845596
------------------------------------------------------------------------------
```

We use `ycen` for the `predict()` option to indicate that we want to predict the censored mean.[5] A random person, actually giving or not, is expected to increase her donation by 18 cents when her income increases by $1,000. Notice that the OLS slope estimator is biased and inconsistent for the partial effect on the censored mean.[6]

Figure 2 illustrates the relationship between the partial effect on the censored mean and income. To produce this figure, run the following code after the tobit estimation:

```
. quiet margins, dydx(income) predict(ycen) at(income = (0(2)70))
. marginsplot, noci plotopts(msymbol(none)) plotregion(margin(zero)) title("")
> ytitle("Marginal effects on E(Donations)")
  Variables that uniquely identify margins: income
```

5. `ycen` is the default prediction option in `nehurdle`, meaning that you can suppress `predict(ycen)` in the code above and you would get the same results.
6. The PEM estimate is 0.2440, with a 99% confidence interval from 0.2417 to 0.2458. This shows that OLS is also a biased and inconsistent estimator of the PEM.

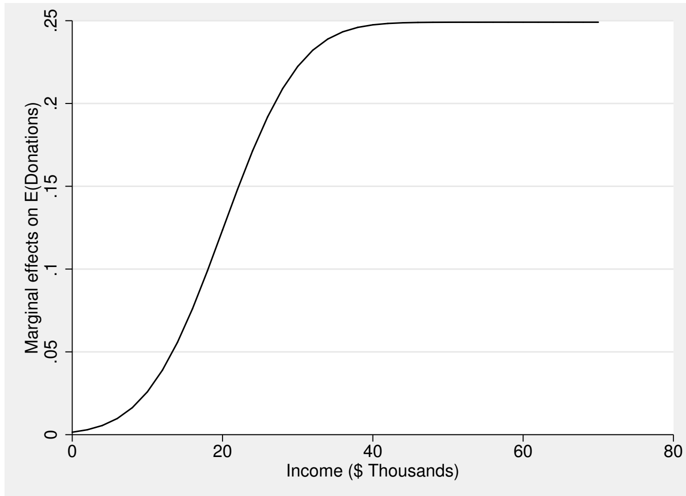

Figure 2. Tobit's partial effects on censored mean

Because we modeled income having a positive effect on giving, people with lower incomes would be less likely to give; therefore, the partial effect on the expected gift of these people should be smaller. As income increases, the probability that a person gives also increases and, consequently, so does the partial effect on the censored mean. The partial effect should never be higher than 0.25, because that is the rate at which the uncensored mean increases. This is exactly what we observe in figure 2.

Figure 3 extends these observations by plotting some observations of `dstar` against `income`, as well as the predicted uncensored and censored means. To produce this figure, run the following code after the tobit estimation:

```
. predict lat, xbval
. label variable lat "Uncensored Mean"
. predict cen, ycen
. label variable cen "Censored Mean"
. twoway (scatter dstar income  if runiform() < 0.05, msymbol(oh) mcolor(gs12))
>         (line lat income, sort lpattern(solid))
>         (line cen income, sort lpattern(shortdash))
>         if income < 100, legend(ring(0) bplacement(seast) cols(1))
>         ytitle(Donations (data and means)) plotregion(margin(zero))
```

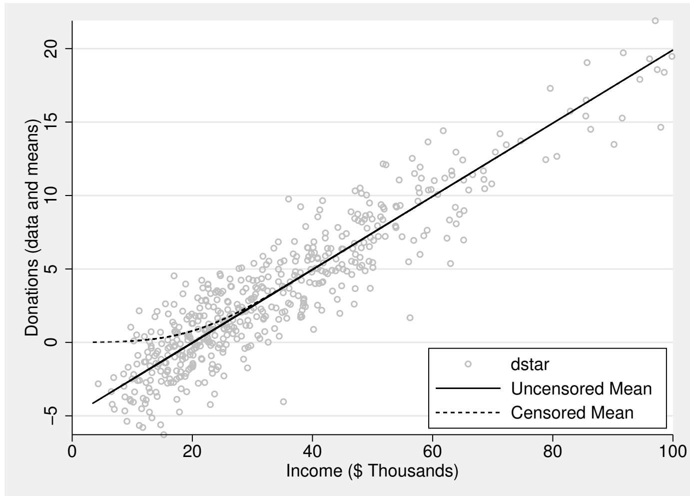

Figure 3. Tobit uncensored and censored means

For low values of income, the censored mean donation is 0 and the curve of the censored mean is almost horizontal, which corresponds to the partial effect at those values being close to 0, as shown in figure 2. As income increases, so does the censored mean and its slope until a point where the censored and uncensored means converge, as do their respective slopes. This is the same level of income at which the partial effect on the censored mean reaches the value of 0.25 in figure 2.

With `nehurdle`, we can also model heteroskedasticity. We do this by adding the `het()` option with the variable list that we want to use to explain it. `nehurdle` models multiplicative heteroskedasticity, so it actually models the natural logarithm of the standard deviation.

The following code fits a linear heteroskedastic tobit model, with the same explanatory variables in the equations for the value and the heteroskedasticity, using the mentioned PSID data:

```
. use psid2005, clear
. local x  "rtincome rexogdon rexogtax spi rpdon i.female npeople i.age
> education"
. nehurdle donation `x´, tobit het(`x´) nolog
Tobit                                              Number of Obs.    =      2645
                                                   Censored Obs.     =      1323
                                                   Uncensored Obs.   =      1322
                                                   Wald chi2(22)     =   1326.84
                                                   Prob > chi2       =    0.0000
Log Likelihood        =   -12417.394               Pseudo R-squared  =    0.1360
```

donation	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
donation						
rtincome	17.38728	1.876081	9.27	0.000	13.71023	21.06433
rexogdon	-.8803535	.6052642	-1.45	0.146	-2.06665	.3059424
rexogtax	.14716	.0588565	2.50	0.012	.0318034	.2625166
spi	-3438.373	731.779	-4.70	0.000	-4872.633	-2004.112
rpdon	-2560.515	603.8302	-4.24	0.000	-3744.001	-1377.03
1.female	313.6857	77.75376	4.03	0.000	161.2912	466.0803
npeople	-62.03832	27.75909	-2.23	0.025	-116.4451	-7.631506
age						
29 < Age <..	30.93684	85.26431	0.36	0.717	-136.1781	198.0518
44 < Age <..	109.9987	103.2044	1.07	0.286	-92.27823	312.2757
Age > 64	875.8475	113.6892	7.70	0.000	653.0208	1098.674
education	63.32247	16.1972	3.91	0.000	31.57654	95.0684
_cons	4196.112	1314.462	3.19	0.001	1619.813	6772.411
lnsigma						
rtincome	.0010712	.0006963	1.54	0.124	-.0002936	.0024359
rexogdon	.0007944	.0004057	1.96	0.050	-6.94e-07	.0015895
rexogtax	-.0001975	.0000389	-5.08	0.000	-.0002737	-.0001213
spi	.1325833	.3137442	0.42	0.673	-.482344	.7475107
rpdon	-1.314431	.2534776	-5.19	0.000	-1.811239	-.8176244
1.female	-.0872354	.0470167	-1.86	0.064	-.1793865	.0049157
npeople	.0818462	.0197774	4.14	0.000	.0430832	.1206092
age						
29 < Age <..	.1486914	.0601236	2.47	0.013	.0308512	.2665316
44 < Age <..	.6023324	.0594146	10.14	0.000	.485882	.7187828
Age > 64	.4206115	.0742798	5.66	0.000	.2750257	.5661973
education	.0897237	.0090379	9.93	0.000	.0720098	.1074375
_cons	7.284881	.5404257	13.48	0.000	6.225666	8.344096

The results present the coefficients of the value equation first and the coefficients of the natural logarithm of the standard deviation equation second.

To get an estimate of the actual standard deviation, not the natural logarithm of the standard deviation, we use `margins` and set `sigma` as the option to `predict` to get an estimate of about $1,642.

```
. margins, predict(sigma)
Predictive margins                              Number of obs     =      2,645
Model VCE    : OIM
Expression   : Prediction of SE, predict(sigma)
```

	Margin	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
_cons	1642.301	35.24999	46.59	0.000	1573.212	1711.39

The APEs for the censored mean are

```
. margins, dydx(*) predict(ycen)
Average marginal effects                        Number of obs     =      2,645
Model VCE    : OIM
Expression   : Prediction of censored mean E(donation|x), predict(ycen)
dy/dx w.r.t. : rtincome rexogdon rexogtax spi rpdon 1.female npeople 1.age
               2.age 3.age education
```

	dy/dx	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
rtincome	7.982701	.8983813	8.89	0.000	6.221906	9.743496
rexogdon	.1045147	.2623958	0.40	0.690	-.4097716	.6188011
rexogtax	-.0562671	.0266987	-2.11	0.035	-.1085955	-.0039386
spi	-1372.146	299.2063	-4.59	0.000	-1958.579	-785.7119
rpdon	-1868.919	239.2978	-7.81	0.000	-2337.934	-1399.904
1.female	75.92467	31.59225	2.40	0.016	14.00501	137.8443
npeople	22.86987	11.40336	2.01	0.045	.519707	45.22004
age						
29 < Age <..	78.65804	29.89365	2.63	0.009	20.06758	137.2485
44 < Age <..	409.365	44.00586	9.30	0.000	323.1151	495.6149
Age > 64	662.5246	61.37603	10.79	0.000	542.2298	782.8194
education	80.51702	6.005978	13.41	0.000	68.74552	92.28852

```
Note: dy/dx for factor levels is the discrete change from the base level.
```

We can also get the APEs on the standard deviation by using `sigma` instead of `ycen` as the option to `predict`:

```
. margins, dydx(*) predict(sigma)
Average marginal effects                        Number of obs     =      2,645
Model VCE    : OIM
Expression   : Prediction of SE, predict(sigma)
dy/dx w.r.t. : rtincome rexogdon rexogtax spi rpdon 1.female npeople 1.age
               2.age 3.age education
```

	dy/dx	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
rtincome	1.759193	1.143025	1.54	0.124	-.4810959	3.999482
rexogdon	1.304628	.6678149	1.95	0.051	-.0042647	2.613522
rexogtax	-.3243744	.0647473	-5.01	0.000	-.4512767	-.1974721
spi	217.7418	515.3677	0.42	0.673	-792.3604	1227.844
rpdon	-2158.692	421.2124	-5.12	0.000	-2984.253	-1333.131
1.female	-145.1859	79.67564	-1.82	0.068	-301.3473	10.97549
npeople	134.4161	33.06077	4.07	0.000	69.61815	199.214
age						
29 < Age <..	194.8683	78.38173	2.49	0.013	41.24295	348.4937
44 < Age <..	1004.486	100.4896	10.00	0.000	807.5302	1201.442
Age > 64	635.5942	122.2269	5.20	0.000	396.034	875.1545
education	147.3533	14.81905	9.94	0.000	118.3085	176.3981

```
Note: dy/dx for factor levels is the discrete change from the base level.
```

3.2 The truncated hurdle model

The truncated hurdle model, also sometimes called two-part model, assumes that there are two distinct and independent processes. One process determines if you are willing to participate in the activity, and this is called the selection, or participation, process. The other process determines how much you are willing to spend on the good, and this is called the value process. The two processes are independent, and to observe a positive amount, you must be both willing to participate and willing to spend a positive amount. This model was first proposed by Cragg (1971) as an extension to the tobit model because it allows much more flexibility to that model.

Let us assume that whether you are willing to give or not is determined by your altruism, A_i^*. We do not observe altruism, but a person will only give if altruism is greater than an unknown value L. Though the value of L is unknown, it is of no consequence because we know that those who give have an altruism greater than L. In addition, the giver will need to have enough income to give so that we observe a positive value. We, therefore, observe

$$D_i = \begin{cases} 0 & \text{if } A_i^* \leq L \text{ or } D_i^* \leq 0 \text{ or both} \\ D_i^* & \text{otherwise} \end{cases}$$

For our example, we continue to assume that uncensored donations are determined by the process in (1). In addition, we assume that the level of altruism is determined by the age of the individual, because we observe that the share of people who give increases with age. So we assume that

$$A_i^* = \delta_0 + \delta_1 \text{age}_i + v_i \tag{2}$$

where $v_i \sim N(0, \sigma_v^2)$. Observing a positive value now depends on both age and income, because age will affect the level of altruism and income will affect the value of the uncensored donation. Our assumptions illustrate the flexibility of the truncated hurdle model relative to the tobit model, in that the truncated hurdle model allows for the selection process to depend on different variables than those that the value process depends on. Only when $A_i^* = D_i^*$ and $L = 0$ is the truncated hurdle model the same as the tobit model. The following code generates the data for this case:

```
. set obs 10000
number of observations (_N) was 0, now 10,000
. set seed 14051969
. generate double income = exp(rnormal(10.31,0.60))/1000
. label variable income "Income ($ Thousands)"
. generate double age = floor((65-18)*runiform() + 18)
. generate double dstar = -5 + 0.25 * income + rnormal(0,2)
. generate double altruism = -2 + 0.08 * age + rnormal(0,1)
. generate dy = altruism > 0 & dstar > 0
. generate double donations = dy * dstar
. summarize age income dy dstar donations
```

Variable	Obs	Mean	Std. Dev.	Min	Max
age	10,000	41.0923	13.54151	18	64
income	10,000	36.25432	24.54682	3.439717	299.9878
dy	10,000	.5848	.4927811	0	1
dstar	10,000	4.050125	6.431671	-9.389664	70.90004
donations	10,000	3.589245	5.585416	0	70.90004

We set the number of observations to 10,000 and generate `income` and `dstar` in the same way we did before. We generate `age` assuming that it is uniformly distributed between 18 and 65, and then we generate altruism by assuming that newborns have an average altruism of -2, that each additional year increases altruism by 0.08, and that the variance in that process is 1. A person will only give if his or her level of altruism is positive, so we have set $L = 0$ in this case. In addition, a person will only give if income is high enough to turn on a positive donation. Thus, we generate `dy` to have a value of 1 if these two conditions are met and 0 otherwise, and then we use it to censor `dstar`.[7] The average of `dy` tells us that $0.5848 \times 10{,}000 = 5{,}848$ observations give donations, so $10{,}000 - 5{,}848 = 4{,}152$ observations are censored.

7. When including a constant in the selection process, setting $L = 0$ is only natural because the constant would capture the actual level of the hurdle point.

The truncated hurdle model is `nehurdle`'s default estimator, so in this case, we do not need to specify an estimator option. The following code estimates the parameters of the model:[8]

```
. nehurdle donations income, select(age income) nolog
Truncated Hurdle                                 Number of Obs.    =     10000
                                                 Censored Obs.     =      4152
                                                 Uncensored Obs.   =      5848
                                                 Wald chi2(3)      =  44370.19
                                                 Prob > chi2       =    0.0000
Log Likelihood       =   -16717.067              Pseudo R-squared  =    0.6654

   donations       Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]

selection
         age    .0360425   .0010656    33.82   0.000     .0339539    .0381311
      income    .0247137   .0007069    34.96   0.000     .0233281    .0260993
       _cons   -2.085337   .0548273   -38.03   0.000    -2.192796   -1.977877

donations
      income    .2492447   .0012097   206.03   0.000     .2468736    .2516157
       _cons   -5.007724   .0686199   -72.98   0.000    -5.142216   -4.873231

    /lnsigma    .7061095   .0107312    65.80   0.000     .6850767    .7271422

       sigma    2.026093   .0217424                      1.983924    2.069159
```

We need to use the `select()` option with the explanatory variables of the selection equation because they differ from those in the value equation. We see that the estimates in the donations (value) equation are very close to the parameters we used when generating `dstar`. Because whether a person gives or not now depends on both processes, we cannot really compare the estimates on the selection equation to the parameters of either of the two processes directly. Like with the tobit model, we can get the APEs on the censored mean by using `margins`:

```
. margins, dydx(*) predict(ycen)
Average marginal effects                         Number of obs     =     10,000
Model VCE    : OIM
Expression   : Prediction of censored mean E(donations|x,z), predict(ycen)
dy/dx w.r.t. : age income

                          Delta-method
                    dy/dx   Std. Err.      z    P>|z|     [95% Conf. Interval]

         age    .0448147   .0012001    37.34   0.000     .0424626    .0471669
      income    .1364251   .0010346   131.86   0.000     .1343973    .1384529
```

8. The following `churdle` command generates the same results: `churdle linear donations income, select(age income) ll(0) nolog`.

Even though age is not part of the uncensored mean equation, it still affects the censored mean. This is because it increases the probability of giving, so it will increase the expected gift of a random person.

To further illustrate `nehurdle`'s functionality, here is how to fit an exponential truncated hurdle model with heteroskedasticity in the selection equation:[9]

9. You can also fit this model with
`churdle exponential donation `x´, select(`x´, het(`xshet´)) ll(0) nolog`.
There is a difference in the value of the log likelihood, which is most likely due to the specification of the log likelihood. `nehurdle`'s log likelihood for the exponential truncated hurdle is specified according to equation 17.55 in Wooldridge (2010). Using `churdle` to fit the exponential (lognormal) truncated hurdle model in table 17.2 of the referred book returns a different log likelihood than the one there, whereas `nehurdle` returns the same one.

```
. use psid2005, clear
. local x  "rtincome rexogdon rexogtax spi rpdon i.female npeople i.age education"
. local xshet "rtincome i.female npeople education"
. nehurdle donation `x´, select(, het(`xshet´)) exponential nolog
Exponential Truncated Hurdle                       Number of Obs.    =      2645
                                                   Censored Obs.     =      1323
                                                   Uncensored Obs.   =      1322
                                                   Wald chi2(26)     =   1031.01
                                                   Prob > chi2       =    0.0000
Log Likelihood       =   -11712.618                Pseudo R-squared  =    0.1022
```

donation	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
selection						
rtincome	.0140792	.004098	3.44	0.001	.0060474	.0221111
rexogdon	-.0004666	.0003992	-1.17	0.243	-.001249	.0003159
rexogtax	.0000327	.0000379	0.86	0.388	-.0000416	.0001071
spi	-1.144201	.5466929	-2.09	0.036	-2.2157	-.0727027
rpdon	-1.382404	.5400135	-2.56	0.010	-2.440811	-.3239968
1.female	.2183704	.0725706	3.01	0.003	.0761346	.3606063
npeople	-.0267615	.0192678	-1.39	0.165	-.0645257	.0110026
age						
29 < Age <..	.0743054	.056626	1.31	0.189	-.0366797	.1852904
44 < Age <..	.2765637	.0926958	2.98	0.003	.0948833	.4582441
Age > 64	.6824733	.1930683	3.53	0.000	.3040663	1.06088
education	.0831606	.0227571	3.65	0.000	.0385576	.1277637
_cons	.953608	.8726205	1.09	0.274	-.7566967	2.663913
lndonation						
rtincome	.0065368	.0014508	4.51	0.000	.0036933	.0093803
rexogdon	.0015578	.0006961	2.24	0.025	.0001936	.002922
rexogtax	-.0001341	.0000673	-1.99	0.046	-.0002659	-2.18e-06
spi	-2.888145	.5947449	-4.86	0.000	-4.053823	-1.722466
rpdon	-2.905677	.4492154	-6.47	0.000	-3.786122	-2.025231
1.female	.137962	.0847014	1.63	0.103	-.0280497	.3039737
npeople	.0269047	.029449	0.91	0.361	-.0308142	.0846236
age						
29 < Age <..	.4395722	.1092145	4.02	0.000	.2255158	.6536287
44 < Age <..	.7089277	.1070452	6.62	0.000	.499123	.9187324
Age > 64	1.344861	.1325245	10.15	0.000	1.085117	1.604604
education	.1374524	.0179718	7.65	0.000	.1022283	.1726766
_cons	8.816798	1.049992	8.40	0.000	6.758852	10.87474
sellnsigma						
rtincome	.0063115	.0016365	3.86	0.000	.0031041	.009519
1.female	-.2066101	.1074848	-1.92	0.055	-.4172765	.0040563
npeople	.0670974	.0355553	1.89	0.059	-.0025897	.1367844
education	-.0309549	.0184875	-1.67	0.094	-.0671898	.00528
/lnsigma	.3200272	.0194477	16.46	0.000	.2819104	.3581441
sigma	1.377165	.0267828			1.32566	1.430672

```
. margins, predict(selsigma)

Predictive margins                              Number of obs     =      2,645
Model VCE    : OIM

Expression   : Prediction of selection SE, predict(selsigma)
```

	Margin	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
_cons	.8271923	.2192823	3.77	0.000	.3974069	1.256978

By specifying the `exponential` option, `nehurdle` models the natural logarithm of the explained variable in the value equation. To specify that we want to model heteroskedasticity in the selection equation, we specify the `het()` option with the explanatory variables for the heteroskedasticity within the `select()` option. Notice that since we are using the same explanatory variables for the selection equation as for the value equation, we do not need to list them in the `select()` option. Just like with the standard deviation of the value equation, we can estimate the standard deviation of the selection equation by using `margins` with `selsigma` as the option to `predict`. Like before, we can get the APEs on the censored mean and on the standard deviation of the selection equation.

```
. margins, dydx(*) predict(ycen)

Average marginal effects                        Number of obs     =      2,645
Model VCE    : OIM

Expression   : Prediction of censored mean E(donation|x,z), predict(ycen)
dy/dx w.r.t. : rtincome rexogdon rexogtax spi rpdon 1.female npeople 1.age
               2.age 3.age education
```

	dy/dx	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
rtincome	8.992593	1.243463	7.23	0.000	6.55545	11.42974
rexogdon	.8761249	.4961328	1.77	0.077	-.0962775	1.848527
rexogtax	-.078112	.0478399	-1.63	0.103	-.1718765	.0156524
spi	-2360.444	466.1509	-5.06	0.000	-3274.083	-1446.805
rpdon	-2459.5	377.1618	-6.52	0.000	-3198.724	-1720.277
1.female	182.6659	57.34457	3.19	0.001	70.27265	295.0592
npeople	2.332226	21.32152	0.11	0.913	-39.45718	44.12163
age						
29 < Age <..	200.8992	48.67793	4.13	0.000	105.4923	296.3062
44 < Age <..	448.595	63.26526	7.09	0.000	324.5974	572.5927
Age > 64	1423.257	208.1354	6.84	0.000	1015.319	1831.195
education	125.5988	15.62074	8.04	0.000	94.98267	156.2149

```
Note: dy/dx for factor levels is the discrete change from the base level.
```

```
. margins, dydx(`xshet´) predict(selsigma)
Average marginal effects                        Number of obs     =      2,645
Model VCE    : OIM
Expression   : Prediction of selection SE, predict(selsigma)
dy/dx w.r.t. : rtincome 1.female npeople education
```

	dy/dx	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
rtincome	.0052208	.0018694	2.79	0.005	.001557	.0088847
1.female	-.1759024	.0954768	-1.84	0.065	-.3630335	.0112286
npeople	.0555024	.0391569	1.42	0.156	-.0212437	.1322485
education	-.0256056	.0097385	-2.63	0.009	-.0446927	-.0065186

```
Note: dy/dx for factor levels is the discrete change from the base level.
```

3.3 Type II tobit model

The type II tobit model assumes that the selection and value processes are different, like with the truncated hurdle model, but not independent. In our case, altruism and donations are not independent. We keep assuming that uncensored donations are still generated by the process in (1) and that altruism is generated by the process in (2). We further assume that the unobserved characteristics of the uncensored donations, u_i, and the unobserved characteristics of altruism, v_i, are not independent from each other. They are, in fact, correlated by a constant factor ρ. In other words, u and v follow a bivariate normal distribution:

$$(u, v) \sim N_2(0, 0, \sigma_u, \sigma_v, \rho)$$

This presents a problem when using a linear specification of the value equation. Because both altruism and donations are jointly determined, whether the person presents a positive donation should be determined in that joint process. There is no guarantee, however, that this is the case. We could have cases where altruism is positive and the person is willing to donate, but we still have negative uncensored donations because the person's income is not large enough. This is illustrated in the following data generation:

```
. clear all
. set obs 10000
number of observations (_N) was 0, now 10,000
. set seed 14051969
. generate double income = exp(rnormal(10.31,0.60))/1000
. generate double age = floor((65-18)*runiform() + 18)
. label var income "Income ($ Thousands)"
. // Altruism sd = 1, dstar sd = 2, and correlation = 0.6
. // Vu = 2^2 = 4, Vv = 1^2 = 1, Cov = 2 * 1 * 0.6 = 1.2
. matrix vcv = (4, 1.2 \ 1.2, 1)
. drawnorm u v, cov(vcv)
. generate double dstar = -5 + 0.25 * income + u
. generate double altruism = -2 + 0.08 * age + v
```

```
. summarize dstar if altruism > 0
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
       dstar |      7,913    4.266464    6.418775  -8.118663   70.90004
```

We generate `income` and `age` in the usual way. To generate the error terms for both processes, selection and value, we use the command `drawnorm`, so we need to create the variance–covariance matrix for the binormal distribution, `vcv`. From before, we have that $\sigma_u = 2$ and $\sigma_v = 1$. Setting the correlation between the error terms to 0.6, then, we have $\sigma_u^2 = 2^2 = 4$ and $\sigma_v^2 = 1^2 = 1$, and the covariance between the two variables is $\sigma_{uv} = \sigma_u \times \sigma_v \times \rho = 2 \times 1 \times 0.6 = 1.2$. Using the `cov()` option of `drawnorm` with `vcv`, we generate the two error terms, `u` and `v`, and then use them to generate `dstar` and `altruism`, respectively. The descriptive statistics of `dstar` for those observations where altruism is positive illustrate the issue. We see that the minimum value is negative, and thus we have negative values even though altruism is positive. We cannot use the type II tobit estimator in this case, and that is why Wooldridge (2010, sec. 17.6.3) explains that we should not use a linear type II tobit model with variables with corner solutions. Only in the case where altruism being positive always implies that the uncensored donation is positive can we use a type II tobit with a linear specification of the value equation. The question is: how can we be sure this is the case when we only observe the censored variable?

This is not a problem for an exponential specification of the value equation, because all uncensored donations will be positive, so overcoming the selection hurdle guarantees that the uncensored mean is positive. I thus present the estimation of an exponential type II tobit using the PSID data with heteroskedasticity in the value equation, to illustrate another `nehurdle` functionality.

```
. use psid2005, clear
. local x "rtincome rexogdon rexogtax spi rpdon i.female npeople i.age education"
. nehurdle donation `x´, heckman exponential het(`x´) nolog

Exponential Type II Tobit                         Number of Obs.    =      2645
                                                  Censored Obs.     =      1323
                                                  Uncensored Obs.   =      1322

                                                  Wald chi2(33)     =    574.78
                                                  Prob > chi2       =    0.0000
Log Likelihood       =   -11713.667               Pseudo R-squared  =    0.1523
```

donation	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
selection						
rtincome	.0128058	.001439	8.90	0.000	.0099855	.0156262
rexogdon	-.0007223	.0004767	-1.52	0.130	-.0016567	.000212
rexogtax	.0000399	.0000466	0.86	0.392	-.0000515	.0001313
spi	-1.077685	.489279	-2.20	0.028	-2.036654	-.1187159
rpdon	-1.38493	.3949017	-3.51	0.000	-2.158924	-.6109372
1.female	.2622548	.0563042	4.66	0.000	.1519007	.372609
npeople	-.0290998	.0174824	-1.66	0.096	-.0633646	.005165
age						
29 < Age <..	.0999744	.0676992	1.48	0.140	-.0327135	.2326624
44 < Age <..	.3263311	.0689974	4.73	0.000	.1910986	.4615636
Age > 64	.8408917	.093307	9.01	0.000	.6580133	1.02377
education	.110891	.0119035	9.32	0.000	.0875605	.1342214
_cons	.5454195	.8878354	0.61	0.539	-1.194706	2.285545
lndonation						
rtincome	.0015173	.0012179	1.25	0.213	-.0008697	.0039043
rexogdon	.0021874	.0007795	2.81	0.005	.0006595	.0037153
rexogtax	-.0001829	.0000748	-2.45	0.014	-.0003295	-.0000364
spi	-1.848296	.6437476	-2.87	0.004	-3.110018	-.5865737
rpdon	-1.831871	.4844596	-3.78	0.000	-2.781395	-.8823479
1.female	-.0092024	.0969034	-0.09	0.924	-.1991296	.1807248
npeople	.0629633	.0354619	1.78	0.076	-.0065408	.1324674
age						
29 < Age <..	.3420478	.1305857	2.62	0.009	.0861045	.5979911
44 < Age <..	.4335339	.1306735	3.32	0.001	.1774185	.6896493
Age > 64	.7850466	.1744053	4.50	0.000	.4432186	1.126875
education	.045011	.0251523	1.79	0.074	-.0042867	.0943087
_cons	9.120548	1.070085	8.52	0.000	7.02322	11.21788
lnsigma						
rtincome	-.0015718	.0007062	-2.23	0.026	-.002956	-.0001876
rexogdon	.0002428	.0003587	0.68	0.498	-.0004601	.0009458
rexogtax	-.0000356	.0000354	-1.01	0.315	-.0001049	.0000338
spi	.8478644	.3054336	2.78	0.006	.2492255	1.446503
rpdon	.5903887	.2343428	2.52	0.012	.1310853	1.049692
1.female	.0075245	.0435758	0.17	0.863	-.0778826	.0929316
npeople	-.0022053	.0161125	-0.14	0.891	-.0337853	.0293746
age						
29 < Age <..	.0354615	.0557482	0.64	0.525	-.073803	.144726
44 < Age <..	-.0354815	.0546721	-0.65	0.516	-.1426368	.0716738
Age > 64	-.1391442	.071796	-1.94	0.053	-.2798618	.0015734
education	-.0168445	.0091462	-1.84	0.066	-.0347707	.0010816
_cons	-.6173745	.5341344	-1.16	0.248	-1.664259	.4295097
/athrho	-.8732372	.1423873	-6.13	0.000	-1.152311	-.5941632
rho	-.7030151	.0720152			-.8185183	-.5328832

```
. margins, predict(sigma)

Predictive margins                              Number of obs     =      2,645
Model VCE    : OIM

Expression   : Prediction of SE, predict(sigma)

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |   1.677411   .0849839    19.74   0.000     1.510846    1.843977
------------------------------------------------------------------------------

. margins, predict(lambda)

Predictive margins                              Number of obs     =      2,645
Model VCE    : OIM

Expression   : Prediction of coefficient on inverse mills ratio,
               predict(lambda)

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |  -1.179245   .1767379    -6.67   0.000    -1.525645   -.8328456
------------------------------------------------------------------------------
```

We see that, to fit a type II tobit model, we need to specify the `heckman` option. The name of the option follows the common naming of this model in the sample selection literature, given that Heckman (1976, 1979) proposed a two-step estimator. This naming is so common that Stata's command to fit sample selection models is named `heckman`. The fact that we are using an estimator that is commonly used to deal with sample selection does not mean that a variable with corner solutions is a case of sample selection—it is not. Having said that, the naming of the option for the type II tobit model should make it easier for Stata users to identify it.

As we have seen before, we use the `exponential` option to specify the exponential value process and the `het()` option to specify the explanatory variables of the multiplicative heteroskedasticity of the value process. We then use `margins` to estimate the standard deviation of the value process by using the `predict(sigma)` option and the coefficient on the inverse Mills ratio, that is, the covariance between both processes, by using the `predict(lambda)` option. Our estimates indicate that the correlation between the two processes is significantly negative. We see that `nehurdle` actually estimates the inverse hyperbolic tangent of the correlation, as does the `heckman` command as well. To test whether the correlation is 0 or not, all we need is the test of individual significance of the inverse hyperbolic tangent of the correlation. The reason is that $\tanh(0) = 0$, so if `/athrho` $= 0$, the correlation will also be 0.

Because we are estimating an exponential type II tobit with heteroskedasticity in the value equation, we can also estimate the APEs on the censored mean and on the standard deviation of the value process.

```
. margins, dydx(*) predict(ycen)
Average marginal effects                          Number of obs     =      2,645
Model VCE    : OIM
Expression   : Prediction of censored mean E(donation|x,z), predict(ycen)
dy/dx w.r.t. : rtincome rexogdon rexogtax spi rpdon 1.female npeople 1.age
               2.age 3.age education

------------------------------------------------------------------------------
             |            Delta-method
             |      dy/dx   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    rtincome |   8.099306   1.144066     7.08   0.000     5.856978    10.34163
    rexogdon |   .8957971   .5189123     1.73   0.084    -.1212522    1.912846
    rexogtax |  -.0967163   .0503641    -1.92   0.055    -.1954281    .0019956
         spi |  -1228.693   419.9808    -2.93   0.003    -2051.841    -405.546
       rpdon |   -1562.75   322.2811    -4.85   0.000    -2194.409   -931.0904
    1.female |   158.8424   56.09592     2.83   0.005     48.89643    268.7884
     npeople |   15.22604   23.58817     0.65   0.519    -31.00593    61.45801
             |
         age |
29 < Age <.. |   185.5922   56.98686     3.26   0.001     73.89998    297.2843
44 < Age <.. |   344.4277   62.90572     5.48   0.000     221.1348    467.7206
    Age > 64 |   1057.336      183.7     5.76   0.000     697.2906    1417.382
             |
   education |   86.01077   13.83128     6.22   0.000     58.90196    113.1196
------------------------------------------------------------------------------
Note: dy/dx for factor levels is the discrete change from the base level.
. margins, dydx(*) predict(sigma)
Average marginal effects                          Number of obs     =      2,645
Model VCE    : OIM
Expression   : Prediction of SE, predict(sigma)
dy/dx w.r.t. : rtincome rexogdon rexogtax spi rpdon 1.female npeople 1.age
               2.age 3.age education

------------------------------------------------------------------------------
             |            Delta-method
             |      dy/dx   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    rtincome |  -.0026365   .0012358    -2.13   0.033    -.0050587   -.0002143
    rexogdon |   .0004073   .0006024     0.68   0.499    -.0007735    .0015881
    rexogtax |  -.0000596   .0000595    -1.00   0.316    -.0001762    .0000569
         spi |   1.422217   .5215573     2.73   0.006     .3999836    2.444451
       rpdon |   .9903246   .4019657     2.46   0.014     .2024862    1.778163
    1.female |   .0126059   .0728087     0.17   0.863    -.1300966    .1553083
     npeople |  -.0036993   .0270215    -0.14   0.891    -.0566604    .0492618
             |
         age |
29 < Age <.. |   .0615684   .0964484     0.64   0.523    -.1274671    .2506038
44 < Age <.. |  -.0594563   .0923021    -0.64   0.519    -.2403651    .1214526
    Age > 64 |  -.2215533   .1160094    -1.91   0.056    -.4489275    .0058209
             |
   education |  -.0282552    .015728    -1.80   0.072    -.0590814     .002571
------------------------------------------------------------------------------
Note: dy/dx for factor levels is the discrete change from the base level.
```

Clearly, you can also model heteroskedasticity in the selection equation in the same way we did in the estimation of the exponential truncated hurdle model.

In addition, and for completeness, `nehurdle` does allow for the estimation of a type II tobit model with a linear specification of the value function, but always keep in mind the previous discussion of the problems with that specification of the model.

4 Conclusions

In this article, I have introduced an estimation command, `nehurdle`, that estimates via maximum likelihood the three popular models used with variables with corner solutions: the tobit model, the truncated hurdle model, and the type II tobit model. I have explained the assumptions behind each estimator not only from a descriptive side, but also from a practical one by generating datasets for the homoskedastic linear specification of each model that follow their assumptions. In addition, I illustrated how to model heteroskedasticity in the value equation, and in the selection equation where appropriate, and to specify an exponential value equation. Furthermore, I illustrated with the use of `margins` how to get estimates of the partial effects on the censored mean and of the standard deviation of the appropriate equation when modeling heteroskedasticity. The partial effects on the censored mean are of particular importance with variables with corner solutions because what we actually observe is the censored variable, not the uncensored one.

Because the purpose of this article is to introduce the estimation command, I have only considered `nehurdle`'s most important functionality. `nehurdle` is a `byable` command; that is, it can be used with the `by` prefix to do the same estimation across different groups or categories. It is also designed to be used with advanced survey techniques with the `svy` prefix. It provides other `predict` options than those presented here; the other options are documented in its help files.

5 References

Cragg, J. G. 1971. Some statistical models for limited dependent variables with application to the demand for durable goods. *Econometrica* 39: 829–844.

Heckman, J. J. 1976. The common structure of statistical models of truncation, sample selection, and limited dependent variables and a simple estimator for such models. *Annals of Economic and Social Measurement* 5: 120–137. http://www.nber.org/chapters/c10494.

———. 1979. Sample selection bias as a specification error. *Econometrica* 47: 153–161.

Tobin, J. 1958. Estimation of relationships for limited dependent variables. *Econometrica* 26: 24–36.

Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

About the author

Alfonso Sánchez-Peñalver has a PhD in econometrics and international economics from Suffolk University. He was born in Madrid, Spain, and has lived in several different countries (England, Switzerland, France, and the United States). He loves soccer with a passion, and he loves playing around with Stata.