# Title

> **churdle** — Cragg hurdle regression

| | | | |
|---|---|---|---|
| Description | Quick start | Menu | Syntax |
| Options | Remarks and examples | Stored results | Methods and formulas |
| References | Also see | | |

## Description

churdle fits a linear or exponential hurdle model for a bounded dependent variable. The hurdle model combines a selection model that determines the boundary points of the dependent variable with an outcome model that determines its nonbounded values. Separate independent covariates are permitted for each model.

## Quick start

Linear hurdle model of y1 on x1 and x2, specifying that y1 is truncated at 0 with x1 and x3 predicting selection

```
churdle linear y1 x1 x2, select(x1 x3) ll(0)
```

Add an upper truncation limit of 40

```
churdle linear y1 x1 x2, select(x1 x3) ll(0) ul(40)
```

Same as above, with the upper truncation limit specified in trunc

```
churdle linear y1 x1 x2, select(x1 x3) ll(0) ul(trunc)
```

Same as above, and use x3 to model the variance of the selection model

```
churdle linear y1 x1 x2, select(x1 x3, het(x3)) ll(0) ul(trunc)
```

Same as above, and use x4 to model the variance of the outcome model

```
churdle linear y1 x1 x2, select(x1 x3, het(x3)) ll(0)     ///
        ul(trunc) het(x4)
```

Exponential hurdle model of y2 on x1 and x2, specifying that y2 is truncated at 4 with x1 and x3 predicting selection

```
churdle exponential y2 x1 x2, select(x1 x3) ll(4)
```

## Menu

Statistics > Linear models and related > Hurdle regression

## Syntax

*Basic syntax*

> churdle <u>lin</u>ear *depvar*, <u>sel</u>ect(*varlist$_s$*) {ll(...)|ul(...)}

> churdle <u>exp</u>onential *depvar*, <u>sel</u>ect(*varlist$_s$*) ll(...)

*Full syntax for churdle linear*

> churdle <u>lin</u>ear *depvar* [*indepvars*] [*if*] [*in*] [*weight*],
>
>      <u>sel</u>ect(*varlist$_s$*[, <u>nocons</u>tant het(*varlist$_o$*)])
>
>      {ll(#|*varname*) | ul(#|*varname*)} [*options*]

*Full syntax for churdle exponential*

> churdle <u>exp</u>onential *depvar* [*indepvars*] [*if*] [*in*] [*weight*],
>
>      <u>sel</u>ect(*varlist$_s$*[, <u>nocons</u>tant het(*varlist$_o$*)]) ll(#|*varname*) [*options*]

| *options* | Description |
|---|---|
| **[Model]** | |
| * <u>sel</u>ect() | specify independent variables and options for selection model |
| ‡ ll(#\|*varname*) | lower truncation limit |
| ‡ ul(#\|*varname*) | upper truncation limit |
| <u>nocons</u>tant | suppress constant term |
| <u>constraints</u>(*constraints*) | apply specified linear constraints |
| het(*varlist*) | specify variables to model the variance |
| **[SE/Robust]** | |
| vce(*vcetype*) | *vcetype* may be oim, <u>r</u>obust, <u>cl</u>uster *clustvar*, <u>boot</u>strap, or <u>jack</u>knife |
| **[Reporting]** | |
| <u>l</u>evel(#) | set confidence level; default is level(95) |
| <u>nocns</u>report | do not display constraints |
| *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; seldom used |
| <u>coefl</u>egend | display legend instead of statistics |

∗select() is required.

    The full specification is <u>select</u>(*varlist$_s$* $\big[$ , <u>nocons</u>tant het(*varlist$_o$*) $\big]$ ).

    noconstant specifies that the constant be excluded from the selection model.

    het(*varlist$_o$*) specifies the variables in the error-variance function of the selection model.

‡You must specify at least one of ul(# $|$ *varname*) or ll(# $|$ *varname*) for the linear model and must specify
    ll(# $|$ *varname*) for the exponential model.

*indepvars*, *varlist$_s$*, and *varlist$_o$* may contain factor variables; see [U] **11.4.3 Factor variables**.

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

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

vce() and weights are not allowed with the svy prefix; see [SVY] **svy**.

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

<u>select</u>(*varlist$_s$* $\big[$ , <u>nocons</u>tant het(*varlist$_o$*) $\big]$ ) specifies the variables and options for the selection
    model. select() is required.

ll(# $|$ *varname*) and ul(# $|$ *varname*) indicate the lower and upper limits, respectively, for the
    dependent variable. You must specify one or both for the linear model and must specify a lower
    limit for the exponential model. Observations with *depvar* $\leq$ ll() have a lower bound; observations
    with *depvar* $\geq$ ul() have an upper bound; and the remaining observations are in the continuous
    region.

noconstant, constraints(*constraints*); see [R] **Estimation options**.

het(*varlist*) specifies the variables in the error-variance function of the outcome model.

      SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived
    from asymptotic theory (oim), 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*.

      Reporting

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

*display_options*: noci, <u>nop</u>values, <u>noo</u>mitted, vsquish, noemptycells, <u>base</u>levels,
    allbaselevels, <u>nofvl</u>abel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*),
    sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

      Maximization

*maximize_options*: <u>di</u>fficult, <u>tech</u>nique(*algorithm_spec*), <u>iter</u>ate(#), $\big[$no$\big]$log, <u>tr</u>ace,
    gradient, showstep, <u>hess</u>ian, <u>showtol</u>erance, <u>tol</u>erance(#), <u>lt</u>olerance(#),
    <u>nrtol</u>erance(#), <u>nonrtol</u>erance, and from(*init_specs*); see [R] **Maximize**. These options are
    seldom used.

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

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

# Remarks and examples                                                    stata.com

`churdle` fits a linear or an exponential hurdle model. It combines a selection model that determines the boundary points of the dependent variable with an outcome model that determines its nonbounded values. Hurdle models treat these boundary values as observed instead of censored. That is to say, observations where the dependent variable is equal to one of the boundary values are not the result of our inability to observe the distribution above or below a certain point; see Wooldridge (2010) chapter 17 for a thorough discussion of this point.

These models were proposed by Cragg (1971) to explain the demand for durable goods. In the Cragg model, individuals purchase zero or a positive amount of the durable good, with different factors determining each of these choices. This may be generalized to other individual decisions, such as money donated to charity, cigarette consumption, and time spent volunteering.

Hurdle models are characterized by the relationship $y_i = s_i h_i^*$, where $y_i$ is the observed value of the dependent variable.

The selection variable, $s_i$, is 1 if the dependent variable is not bounded and 0 otherwise. In the Cragg model, the lower limit that binds the dependent variable is 0 so the selection model is

$$s_i = \begin{cases} 1 & \text{if } \mathbf{z}_i \boldsymbol{\gamma} + \epsilon_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{z}_i$ is a vector of explanatory variables, $\boldsymbol{\gamma}$ is a vector of coefficients, and $\epsilon_i$ is a standard normal error term. `churdle` allows a different lower limit to be specified in `ll()` and, for the linear model, an upper limit in `ul()`. Conditional heteroskedasticity of the random error $\epsilon_i$ is allowed if suboption `het()` is specified in `select()`.

The continuous latent variable $h_i^*$ is observed only if $s_i = 1$. The outcome model can be either the linear model or the exponential model, as proposed in Cragg (1971):

$$h_i^* = \mathbf{x}_i \boldsymbol{\beta} + \nu_i \qquad \text{(linear)}$$
$$h_i^* = \exp\left(\mathbf{x}_i \boldsymbol{\beta} + \nu_i\right) \quad \text{(exponential)}$$

where $\mathbf{x}_i$ is a vector of explanatory variables, $\boldsymbol{\beta}$ is a vector of coefficients, and $\nu_i$ is an error term.

For the linear model, $\nu_i$ has a truncated normal distribution with lower truncation point $-\mathbf{x}_i \boldsymbol{\beta}$. For the exponential model, $\nu_i$ has a normal distribution. `churdle` extends the Cragg hurdle models to allow for conditional heteroskedasticity of the random error $\nu_i$ if the user specifies the `het()` option.

The parameters and regressors in the models for $h_i^*$ and for $s_i$ may differ.

▷ Example 1: Linear hurdle model

Consider a dataset that contains the number of hours an individual exercises per day (hours), their age (age), whether they are single (single), hours they work per day (whours), whether they smoke (smoke), their weight in kilograms (weight), their distance from the nearest gym (distance), and their average commute from work (commute).
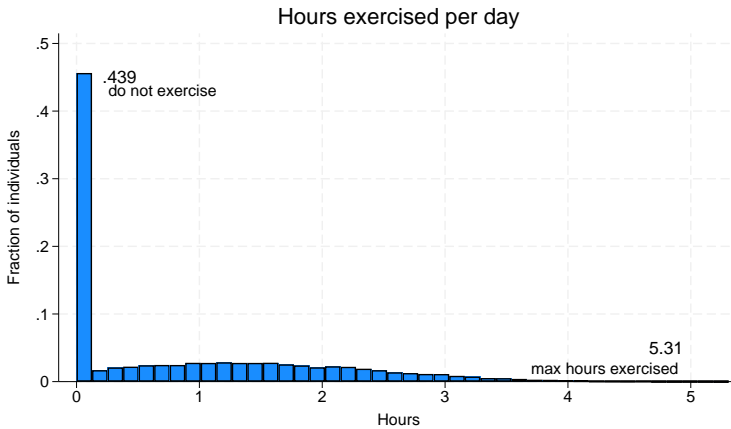


Figure 1.

Figure 1 shows that 43.9% of the individuals in the sample do not exercise and that the hours exercised varies among individuals that decide to exercise.

We model the decision to exercise or not as a function of commute, whours, and age. These variables are written in select(). Once a decision to exercise is made, the time an individual exercises is modeled as a linear function of age, smoke, distance, and single.

```
. use https://www.stata-press.com/data/r18/fitness
(Fictional fitness data)

. churdle linear hours age i.smoke distance i.single,
> select(commute whours age) ll(0)
Iteration 0:  Log likelihood = -23657.236
Iteration 1:  Log likelihood = -23344.182
Iteration 2:  Log likelihood = -23340.051
Iteration 3:  Log likelihood = -23340.044
Iteration 4:  Log likelihood = -23340.044
```

| Cragg hurdle regression | | | | | Number of obs | = 19,831 |
|---|---|---|---|---|---|---|
| | | | | | LR chi2(4) | = 9059.26 |
| | | | | | Prob > chi2 | = 0.0000 |
| Log likelihood = -23340.044 | | | | | Pseudo R2 | = 0.1625 |

| hours | Coefficient | Std. err. | z | P>\|z\| | [95% conf. interval] | |
|---|---|---|---|---|---|---|
| **hours** | | | | | | |
| age | .0015116 | .000763 | 1.98 | 0.048 | .0000162 | .003007 |
| | | | | | | |
| **smoke** | | | | | | |
| Smoking | -1.06646 | .0460578 | -23.15 | 0.000 | -1.156731 | -.9761879 |
| distance | -.1333868 | .0126344 | -10.56 | 0.000 | -.1581497 | -.1086238 |
| | | | | | | |
| **single** | | | | | | |
| Single | .9940893 | .0258775 | 38.42 | 0.000 | .9433703 | 1.044808 |
| _cons | .9138855 | .0396227 | 23.06 | 0.000 | .8362264 | .9915447 |
| **selection_ll** | | | | | | |
| commute | -.2953345 | .0624665 | -4.73 | 0.000 | -.4177666 | -.1729024 |
| whours | .0022974 | .0069306 | 0.33 | 0.740 | -.0112864 | .0158811 |
| age | -.0485347 | .0006501 | -74.65 | 0.000 | -.049809 | -.0472604 |
| _cons | 2.649945 | .0499795 | 53.02 | 0.000 | 2.551987 | 2.747903 |
| **lnsigma** | | | | | | |
| _cons | .0083199 | .0099648 | 0.83 | 0.404 | -.0112107 | .0278506 |
| **/sigma** | 1.008355 | .010048 | | | .9888519 | 1.028242 |

The coefficients in the outcome model for the latent variable appear under `hours`. Because we only specified a lower limit to bind the dependent variable, the output shows parameter estimates for a single selection model under `selection_ll`. Information about the estimated standard deviation of the error term in the outcome model appears under `lnsigma` and `/sigma`.

The coefficient estimates are not directly interpretable. To obtain the effect of a covariate on the model, we need to use the `margins` command; see [R] **churdle postestimation**. Consider the effect of `age`:

```
. margins, dydx(age)
```

Average marginal effects                                            Number of obs = 19,831
Model VCE: OIM

Expression: Conditional mean estimates of dependent variable, predict()
dy/dx wrt:  age

|  | dy/dx | Delta-method std. err. | z | P>\|z\| | [95% conf. interval] | |
|---|---|---|---|---|---|---|
| age | -.0216855 | .000289 | -75.03 | 0.000 | -.022252 | -.021119 |

Each additional year of age is associated with about $-0.02$ fewer hours, or 1.2 minutes, of exercise.
◁

▷ Example 2: Linear hurdle with models for the outcome and selection variances

In this example, we illustrate the possibility of fitting a heteroskedastic probit for the selection and latent model. In both cases, this is done by specifying age and single as the variables that affect the conditional variance. As in example 1, we have separate parameters for the outcome model and lower-limit selection model.

```
. churdle linear hours age i.smoke distance i.single,
> select(commute whours age, het(age single)) ll(0) het(age single) nolog
```

Cragg hurdle regression                                             Number of obs =   19,831
                                                                    LR chi2(4)    =  9060.63
                                                                    Prob > chi2   =   0.0000
Log likelihood = -23339.355                                         Pseudo R2     =   0.1626

| hours | Coefficient | Std. err. | z | P>\|z\| | [95% conf. interval] | |
|---|---|---|---|---|---|---|
| **hours** | | | | | | |
| age | .0012559 | .0008198 | 1.53 | 0.126 | -.0003508 | .0028626 |
| **smoke** | | | | | | |
| Smoking | -1.065564 | .0457657 | -23.28 | 0.000 | -1.155263 | -.9758649 |
| distance | -.1332939 | .0126102 | -10.57 | 0.000 | -.1580094 | -.1085783 |
| **single** | | | | | | |
| Single | 1.002511 | .032535 | 30.81 | 0.000 | .9387436 | 1.066278 |
| _cons | .9166356 | .0388318 | 23.61 | 0.000 | .8405268 | .9927445 |
| **selection_ll** | | | | | | |
| commute | -.2959986 | .0641594 | -4.61 | 0.000 | -.4217488 | -.1702484 |
| whours | .0024514 | .0069769 | 0.35 | 0.725 | -.0112231 | .0161259 |
| age | -.048886 | .0021405 | -22.84 | 0.000 | -.0530814 | -.0446906 |
| _cons | 2.669613 | .1139478 | 23.43 | 0.000 | 2.44628 | 2.892947 |
| **lnsigma** | | | | | | |
| age | .0003537 | .0004026 | 0.88 | 0.380 | -.0004354 | .0011427 |
| single | -.0080667 | .019253 | -0.42 | 0.675 | -.0458019 | .0296685 |
| **lnsigma_ll** | | | | | | |
| age | -.0002035 | .0008424 | -0.24 | 0.809 | -.0018546 | .0014475 |
| single | .0268271 | .0270133 | 0.99 | 0.321 | -.0261179 | .0797721 |

The coefficients on `age` and `single` have no effect on the conditional variance of the outcome model or on the conditional variance of the selection model. Thus, there is no evidence that the variance depends on age and marital status.

◁

## ▷ Example 3: Exponential hurdle model

Returning to example 1, if we believe that the conditional mean of the latent variable has an exponential form instead of a linear form, we use `churdle exponential`.

```
. churdle exponential hours age i.smoke distance i.single,
> select(commute whours age) ll(0) nolog
```

```
Cragg hurdle regression                          Number of obs =   19,831
                                                 LR chi2(4)    = 8663.21
                                                 Prob > chi2   =   0.0000
Log likelihood = -15666.195                      Pseudo R2     =   0.2166
```

| hours | Coefficient | Std. err. | z | P>\|z\| | [95% conf. interval] | |
|---|---|---|---|---|---|---|
| **hours** | | | | | | |
| age | .0008368 | .0005341 | 1.57 | 0.117 | -.00021 | .0018836 |
| **smoke** | | | | | | |
| Smoking | -.6431348 | .0258509 | -24.88 | 0.000 | -.6938016 | -.592468 |
| distance | -.0772879 | .0079132 | -9.77 | 0.000 | -.0927976 | -.0617783 |
| **single** | | | | | | |
| Single | .5975111 | .016108 | 37.09 | 0.000 | .5659401 | .6290821 |
| _cons | -.0770619 | .0254833 | -3.02 | 0.002 | -.1270082 | -.0271157 |
| **selection_ll** | | | | | | |
| commute | -.2953345 | .0624665 | -4.73 | 0.000 | -.4177666 | -.1729024 |
| whours | .0022974 | .0069306 | 0.33 | 0.740 | -.0112864 | .0158811 |
| age | -.0485347 | .0006501 | -74.65 | 0.000 | -.049809 | -.0472604 |
| _cons | 2.649945 | .0499795 | 53.02 | 0.000 | 2.551987 | 2.747903 |
| **lnsigma** | | | | | | |
| _cons | -.186917 | .0067067 | -27.87 | 0.000 | -.200062 | -.1737721 |
| **/sigma** | .8295126 | .0055633 | | | .81868 | .8404884 |

What was said previously regarding the interpretation of the effects of the different regressors also holds true for `churdle exponential`. We again use `margins` to estimate the effect of `age` on time spent exercising.

```
. margins, dydx(age)
```

```
Average marginal effects                         Number of obs = 19,831
Model VCE: OIM
```

Expression: Conditional mean estimates of dependent variable, predict()
dy/dx wrt:  age

| | dy/dx | Delta-method std. err. | z | P>\|z\| | [95% conf. interval] | |
|---|---|---|---|---|---|---|
| age | -.0245582 | .0004805 | -51.11 | 0.000 | -.0255 | -.0236164 |

With the exponential outcome model of the latent variable, our estimate is that each additional year of age decreases exercise time by about 0.025 hours, or 1.5 minutes.

◁

## Stored results

churdle stores the following in e():

Scalars
| | |
|---|---|
| e(N) | number of observations |
| e(k_eq_model) | number of equations in overall model test |
| e(df_m) | model degrees of freedom |
| e(r2_p) | pseudo-$R^2$ |
| e(chi2) | $\chi^2$ |
| e(ll) | log likelihood |
| e(ll_0) | log likelihood, constant-only model |
| e(N_clust) | number of clusters |
| e(p) | $p$-value for model test |
| e(rank) | rank of e(v) |
| e(converged) | 1 if converged, 0 otherwise |

Macros
| | |
|---|---|
| e(cmd) | churdle |
| e(cmdline) | command as typed |
| e(depvar) | name of dependent variable |
| e(estimator) | linear or exponential |
| e(model) | Linear or Exponential |
| e(wtype) | weight type |
| e(wexp) | weight expression |
| e(title) | title in estimation output |
| e(clustvar) | name of cluster variable |
| e(chi2type) | Wald or LR; type of model $\chi^2$ test |
| e(vce) | *vcetype* specified in vce() |
| e(vcetype) | title used to label Std. err. |
| e(opt) | type of optimization |
| e(which) | max or min; whether optimizer is to perform maximization or minimization |
| e(technique) | maximization technique |
| e(properties) | b V |
| e(predict) | program used to implement predict |
| 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(Cns) | constraints matrix |
| e(ilog) | iteration log |
| e(V) | variance–covariance matrix of the estimators |
| e(V_modelbased) | model-based variance |

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

Let $\ell\ell$ refer to the lower limit and $u\ell$ to the upper limit. Also let the probabilities of being at these limits be given by

$$\Pr\left(y_i = \ell\ell | \mathbf{z}_i\right) = \Phi\left(\ell\ell - \mathbf{z}_i'\boldsymbol{\gamma}_{\ell\ell}\right)$$

$$\Pr\left(y_i = u\ell | \mathbf{z}_i\right) = \Phi\left(\mathbf{z}_i'\boldsymbol{\gamma}_{u\ell} - u\ell\right)$$

where $\mathbf{z}_i$ are the covariates of the selection model for individual $i$, which may be distinct from the covariates $\mathbf{x}_i$ for the latent model; $\boldsymbol{\Phi}$ corresponds to the standard normal cumulative distribution function; $\boldsymbol{\gamma}_{\ell\ell}$ is the parameter vector for the lower-limit selection model; and $\boldsymbol{\gamma}_{u\ell}$ is the parameter vector for the upper-limit selection model.

Under the assumptions that $\nu_i$ has a truncated normal distribution with lower truncation point $\ell\ell - \mathbf{x}_i'\boldsymbol{\beta}$ and upper truncation point $u\ell - \mathbf{x}_i'\boldsymbol{\beta}$ and has a homoskedastic variance, the log-likelihood function is given by

$$\begin{aligned}
\ln\mathbf{L} = \sum_{i=1}^{n} &\left(y_i \leq \ell\ell\right) \log\Phi\left(\ell\ell - \mathbf{z}_i'\boldsymbol{\gamma}_{\ell\ell}\right) + \left(y_i \geq u\ell\right) \log\left\{1 - \Phi\left(u\ell - \mathbf{z}_i'\boldsymbol{\gamma}_{u\ell}\right)\right\} \\
&+ \left(u\ell > y_i > \ell\ell\right)\left[\,\log\left\{\Phi\left(u\ell - \mathbf{z}_i'\boldsymbol{\gamma}_{u\ell}\right) - \Phi\left(\ell\ell - \mathbf{z}_i'\boldsymbol{\gamma}_{\ell\ell}\right)\right\}\right] \\
&- \left(u\ell > y_i > \ell\ell\right)\left[\log\left\{\Phi\left(\frac{u\ell - \mathbf{x}_i'\boldsymbol{\beta}}{\sigma}\right) - \Phi\left(\frac{\ell\ell - \mathbf{x}_i'\boldsymbol{\beta}}{\sigma}\right)\right\}\right] \\
&+ \left(u\ell > y_i > \ell\ell\right)\left[\log\left\{\phi\left(\frac{y_i - \mathbf{x}_i'\boldsymbol{\beta}}{\sigma}\right)\right\} - \log(\sigma)\right]
\end{aligned}$$

Without the homoskedasticity assumption, the heteroskedasticity can be modeled using the form $\sigma^2\left(\mathbf{w}_i\right) = \exp\left(2\mathbf{w}_i'\boldsymbol{\theta}\right)$, where $\mathbf{w}_i$ are the variables that affect the conditional variance of $\nu_i$. The log-likelihood function is obtained by replacing $\sigma$ with $\exp\left(\mathbf{w}_i'\boldsymbol{\theta}\right)$.

The log-likelihood function for the exponential model is given by

$$\begin{aligned}
\ln\mathbf{L} = \sum_{i=1}^{n} &\left(y_i \leq \ell\ell\right) \log\Phi\left(\ell\ell - \mathbf{z}_i'\boldsymbol{\gamma}\right) + \left(y_i > \ell\ell\right)\left[\log\left\{1 - \Phi\left(\ell\ell - \mathbf{z}_i'\boldsymbol{\gamma}\right)\right\}\right] \\
&+ \left(y_i > \ell\ell\right)\left\{\log\left\{\phi\left[\log(y_i - \ell\ell) - \mathbf{x}_i'\boldsymbol{\beta}\right)/\sigma\right]\right\} - \log(\sigma) - \log(y_i - \ell\ell)\right\}
\end{aligned}$$

Analogous to the linear case, we can model heteroskedasticity by $\sigma^2\left(\mathbf{w}_i\right) = \exp\left(2\mathbf{w}_i'\boldsymbol{\theta}\right)$.

Estimation of both of the aforementioned likelihood functions is done by maximum likelihood.

# References

Belotti, F., P. Deb, W. G. Manning, and E. C. Norton. 2015. twopm: Two-part models. *Stata Journal* 15: 3–20.

Cragg, J. G. 1971. Some statistical models for limited dependent variables with application to the demand for durable goods. *Econometrica* 39: 829–844. https://doi.org/10.2307/1909582.

Deb, P., E. C. Norton, and W. G. Manning. 2017. *Health Econometrics Using Stata*. College Station, TX: Stata Press.

Engel, C., and P. G. Moffatt. 2014. dhreg, xtdhreg, and bootdhreg: Commands to implement double-hurdle regression. *Stata Journal* 14: 778–797.

Farbmacher, H. 2011. Estimation of hurdle models for overdispersed count data. *Stata Journal* 11: 82–94.

García, B. 2013. Implementation of a double-hurdle model. *Stata Journal* 13: 776–794.

Gray, L. A., and M. Hernández-Alava. 2018. A command for fitting mixture regression models for bounded dependent variables using the beta distribution. *Stata Journal* 18: 51–75.

Marchenko, Y. V. 2015. Bayesian modeling: Beyond Stata's built-in models. *The Stata Blog: Not Elsewhere Classified*. http://blog.stata.com/2015/05/26/bayesian-modeling-beyond-statas-built-in-models/.

Sánchez-Peñalver, A. 2019. Estimation methods in the presence of corner solutions. *Stata Journal* 19: 87–111.

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

Wulff, J. N. 2019. Generalized two-part fractional regression with cmp. *Stata Journal* 19: 375–389.

## Also see

[R] **churdle postestimation** — Postestimation tools for churdle

[R] **intreg** — Interval regression

[R] **tobit** — Tobit regression

[SVY] **svy estimation** — Estimation commands for survey data

[U] **20 Estimation and postestimation commands**