**Description**

`intreg` fits a linear model with an outcome measured as point data, interval data, left-censored data, or right-censored data. As such, it is a generalization of the model fit by `tobit`.

**Quick start**

Regression on `x1` and `x2` of an interval-measured dependent variable with lower endpoint `y_lower` and upper endpoint `y_upper`  
`intreg y_lower y_upper x1 x2`

With robust standard errors  
`intreg y_lower y_upper x1 x2, vce(robust)`

Model heteroskedasticity in the conditional variance as a function of `x3`  
`intreg y_lower y_upper x1 x2, het(x3)`

Adjust for complex survey design using `svyset` data  
`svy: intreg y_lower y_upper x1 x2`

**Menu**

Statistics > Linear models and related > Censored regression > Interval regression
Syntax

```
intreg depvar1 depvar2 [ indepvars ] [ if ] [ in ] [ weight ] [ , options ]
```

depvar1 and depvar2 should have the following form:

<table>
<thead>
<tr>
<th>Type of data</th>
<th>depvar1</th>
<th>depvar2</th>
</tr>
</thead>
<tbody>
<tr>
<td>point data</td>
<td>a = [a,a]</td>
<td>a</td>
</tr>
<tr>
<td>interval data</td>
<td>[a,b]</td>
<td>a</td>
</tr>
<tr>
<td>left-censored data</td>
<td>(−∞,b]</td>
<td>.</td>
</tr>
<tr>
<td>right-censored data</td>
<td>[a,+∞)</td>
<td>a</td>
</tr>
<tr>
<td>missing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**options**

<table>
<thead>
<tr>
<th>Description</th>
<th>Model</th>
<th>SE/Robust</th>
<th>Reporting</th>
<th>Maximization</th>
</tr>
</thead>
<tbody>
<tr>
<td>suppress constant term</td>
<td>noconstant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>independent variables to model the variance; use noconstant to suppress constant term</td>
<td>het(varlist[, noconstant])</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>include varname in model with coefficient constrained to 1</td>
<td>offset(varname)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>apply specified linear constraints</td>
<td>constraints(constraints)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vcetype may be oim, robust, cluster clustvar, opg, bootstrap, or jackknife</td>
<td>vce(vcetype)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>set confidence level; default is level(95)</td>
<td>level(#)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>do not display constraints</td>
<td>nocnsreport</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling</td>
<td>display_options</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>control the maximization process; seldom used</td>
<td>maximize_options</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>keep collinear variables</td>
<td>collinear</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>display legend instead of statistics</td>
<td>coeflegend</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*indepvars* and *varlist* may contain factor variables; see [U] 11.4.3 Factor variables. *depvar1*, *depvar2*, *indepvars*, and *varlist* may contain time-series operators; see [U] 11.4.4 Time-series varlists. bayes, bootstrap, by, fmm, fp, jackknife, mfp, nestreg, rolling, statsby, stepwise, and svy are allowed; see [U] 11.1.10 Prefix commands. For more details, see [BAYES] bayes: intreg and [FMM] fmm: intreg.

Weights are not allowed with the bootstrap prefix; see [R] bootstrap. aweights are not allowed with the jackknife prefix; see [R] jackknife. vce() and weights are not allowed with the svy prefix; see [SVY] svy. aweights, fweights, iweights, and pweights are allowed; see [U] 11.1.6 weight. collinear and coeflegend do not appear in the dialog box.

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

**Model**

noconstant; see [R] Estimation options.

het(varlist [, noconstant]) specifies that the logarithm of the standard deviation be modeled as a linear combination of varlist. The constant is included unless noconstant is specified.

offset(varname), constraints(constraints); see [R] Estimation options.

**SE/Robust**

vce(vcetype) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim, opg), 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, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(style), cformat(%fmt), pformat(%fmt), sformat(%fmt), and nolstretch; see [R] Estimation options.

**Maximization**

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

Setting the optimization type to technique(bhhh) resets the default vcetype to vce(opg).

The following options are available with intreg but are not shown in the dialog box: collinear, coeflegend; see [R] Estimation options.

Remarks and examples

intreg fits a linear model to an outcome that may be either observed exactly or unobserved but known to fall within some interval. The values of the outcome variable may be observed (point data), unobserved but known to fall within an interval with fixed endpoints (interval-censored data), unobserved but known to fall within an interval that has a fixed upper endpoint (left-censored data), or unobserved but known to fall within an interval that has a fixed lower endpoint (right-censored data). Such censored data arise naturally in many contexts, such as wage data. Often, you know only that, for example, a person’s salary is between $30,000 and $40,000.

The interval regression model fit by intreg is a generalization of the models fit by tobit because it extends censoring beyond left-censored data or right-censored data; see Cameron and Trivedi (2010, 548–550) for additional discussion of these data types. See Wooldridge (2020, sec. 17.4) for an introduction to censored and truncated regression models.
Regardless of the type of censoring, `intreg` requires the outcome to be stored in the dataset as interval data. That is, two dependent variables, `depvar1` and `depvar2`, are used to hold the endpoints of the interval. If the data are left-censored, the lower endpoint is $-\infty$ and is represented by a missing value in `depvar1`. If the data are right-censored, the upper endpoint is $+\infty$ and is represented by a missing value in `depvar2`. Point data are represented by the two endpoints being equal. Truly missing values of the dependent variable must be represented by missing values in both `depvar1` and `depvar2`.

Example 1: Interval regression

`womenwage2.dta` contains the yearly wages of working women in interval form. Women were asked to indicate a category for their yearly income from employment. The categories were $5,000 or less, $5,001–$10,000, ..., $25,001–$30,000, $30,001–$40,000, $40,001–$50,000, and more than $50,000. The lower and upper endpoints of the wage categories (in $1,000s) are recorded in variables `wage1` and `wage2`. Below, we list the first 10 observations in `wage1` and `wage2`.

```
. use https://www.stata-press.com/data/r16/womenwage2
(Wages of women, fictional data)
. list wage1 wage2 in 1/10

   wage1 | wage2
----------+---------
        1. |   .      5
        2. |   5      10
        3. |   5      10
        4. |  10      15
        5. |  15      20
        6. |  20      25
        7. |  25      30
        8. |  30      40
        9. |  40      50
       10. |  50      .
```

We see, for example, that the first respondent made $5,000 or less in a year, that the second respondent made between $5,001 and $10,000 in a year, and so on. The tenth respondent made at least $50,000 a year.

We now fit an interval regression model of women’s wages using social and demographic characteristics as explanatory variables. The variables include the subject’s age (`age`), years of schooling (`school`), job tenure (`tenure`), a dummy for living in a rural area (`rural`), and a dummy for never being married (`nev_mar`).
. intreg wage1 wage2 age c.age#c.age i.nev_mar i.rural school tenure

Fitting constant-only model:
Iteration 0:  log likelihood = -967.24956
Iteration 1:  log likelihood = -967.1368
Iteration 2:  log likelihood = -967.1368

Fitting full model:
Iteration 0:  log likelihood = -856.65324
Iteration 1:  log likelihood = -856.33294
Iteration 2:  log likelihood = -856.33293

Interval regression

|                        | Coef.    | Std. Err. | z      | P>|z|  | [95% Conf. Interval] |
|------------------------|----------|-----------|--------|------|---------------------|
| age                    | .7914438 | .4433604  | 1.79   | 0.074| -.0775265           | 1.660414 |
| c.age#c.age            | -.0132624| .0073028  | -1.82  | 0.069| -.0275757           | .0010509 |
| 1.nev_mar              | -.2075022| .8119581  | -0.26  | 0.798| -1.798911           | 1.383906 |
| 1.rural                | -3.043044| .7757324  | -3.92  | 0.000| -4.563452           | -1.522637 |
| school                 | 1.334721 | .1357873  | 9.83   | 0.000| 1.068583            | 1.600859 |
| tenure                 | .8000664 | .1045077  | 7.66   | 0.000| .5952351            | 1.004898 |
| _cons                  | -12.70238| 6.367117  | -1.99  | 0.046| -25.1817           | -.2230583 |
| /lnsigma               | 1.987823 | .0346543  | 57.36  | 0.000| 1.919902           | 2.055744 |
| sigma                  | 7.299626 | .2529634  | 6.82029| 0.000| 7.1265             | 7.81265  |

Because the conditional mean modeled by interval regression is linear, the coefficients are interpreted the same way they are in ordinary least-squares regression; see [R] regress. For example, residing in a rural area lowers the expected income by $3,043 and each additional year of schooling raises the expected income by $1,335.
Instead of using intervals to record wages, we could treat the outcome as categorical with a higher category corresponding to a higher wage. Here we fit an ordered probit model for wagecat, created based on groups defined by the intervals, by using oprobit (see [R] oprobit) with the same covariates:

```
. oprobit wagecat age c.age#c.age i.nev_mar i.rural school tenure
Iteration 0:  log likelihood = -881.1491
Iteration 1:  log likelihood = -764.31729
Iteration 2:  log likelihood = -763.31191
Iteration 3:  log likelihood = -763.31049
Iteration 4:  log likelihood = -763.31049
Ordered probit regression
Number of obs  =  488
LR chi2(6)     =  235.68
Prob > chi2    =  0.0000
Log likelihood = -763.31049
Pseudo R2      =  0.1337

               | Coef.  Std. Err.     z  P>|z|      [95% Conf. Interval]
-------------|---------:--------:------:---------:------------------------
    wagecat   |         
     age     |  .1674519  .0620333    2.70  0.007     .0458689    .289035   
  c.age#c.age| -.0027983  .0010214   -2.74  0.006    -.0048001    -.0007964
 1.nev_mar   | -.0046417  .1126737   -0.04  0.967    -.225478    .2161946
 1.rural    |  -.5270036  .1100449   -4.79  0.000    -.7426875   -.3113196
  school    |  .2010587  .0201189    9.99  0.000     .1616263    .2404911
   tenure   |  .0989916  .0147887    6.69  0.000     .0700063    .1279777
/cut1       |  2.650637   .8957245    2.95  0.003     .960495    4.340225   
/cut2       |  3.941018   .8979167    4.34  0.000     2.181134    5.700903   
/cut3       |  5.085205   .9065682    5.58  0.000     3.310148    6.860263   
/cut4       |  5.875534   .9120933    6.48  0.000     4.087864    7.663204   
/cut5       |  6.468723   .918117    6.97  0.000     4.669247    8.268199   
/cut6       |  6.922726   .9215455    7.57  0.000     5.11653     8.728922   
/cut7       |  7.34471   .9237628    7.97  0.000     5.534168    9.155252   
/cut8       |  7.963441   .9338881   8.58  0.000     6.133054    9.793828
```

We can directly compare the log likelihoods for the intreg and oprobit models because both likelihoods are discrete. If we had point data in our intreg estimation, the likelihood would be a mixture of discrete and continuous terms, and we could not compare it directly with the oprobit likelihood.

Here the oprobit log likelihood is significantly larger (that is, less negative), so it fits better than the intreg model. The intreg model assumes normality, but the distribution of wages is skewed and definitely nonnormal. Normality is more closely approximated if we model the log of wages.
. generate logwage1 = log(wage1)
(14 missing values generated)
. generate logwage2 = log(wage2)
(6 missing values generated)
. intreg logwage1 logwage2 age c.age#c.age i.nev_mar i.rural school tenure

Fitting constant-only model:
Iteration 0:  log likelihood = -889.23647
Iteration 1:  log likelihood = -889.06346
Iteration 2:  log likelihood = -889.06346

Fitting full model:
Iteration 0:  log likelihood = -773.81968
Iteration 1:  log likelihood = -773.36566
Iteration 2:  log likelihood = -773.36563

Interval regression

|            | Coef. | Std. Err. | z   | P>|z| | [95% Conf. Interval] |
|------------|-------|-----------|-----|------|----------------------|
| age        | 0.0645589 | 0.0249954 | 2.58 | 0.010 | 0.0155689 - 0.1135489 |
| c.age#c.age| -0.0010812 | 0.0004115 | -2.63 | 0.009 | -0.0018878 - 0.0002746 |
| 1.nev_mar  | -0.0058151 | 0.0454867 | -0.13 | 0.898 | -0.0949674 - 0.0833371 |
| 1.rural    | -0.2098361 | 0.0439454 | -4.77 | 0.000 | -0.2959675 - 0.1237047 |
| school     | 0.0804832 | 0.0076783 | 10.48 | 0.000 | 0.0654341 - 0.0955323 |
| tenure     | 0.0397144 | 0.0058001 | 6.85 | 0.000 | 0.0283464 - 0.0510825 |
| _cons      | 0.7084023 | 0.3593193 | 1.97 | 0.049 | 0.0041495 - 1.412655 |
| /lnsigma   | -0.906989 | 0.0356265 | -25.46 | 0.000 | -0.9768157 - 0.8371623 |
| sigma      | 0.4037381 | 0.0143838 | 28.15 | 0.000 | 0.3765081 - 0.4329373 |

The log likelihood of this intreg model is close to the oprobit log likelihood, and the z statistics for both models are similar.
Stored results

`intreg` stores the following in `e()`:

Scalars

- `e(N)` number of observations
- `e(N_unc)` number of uncensored observations
- `e(N_lc)` number of left-censored observations
- `e(N_rc)` number of right-censored observations
- `e(N_int)` number of interval observations
- `e(k)` number of parameters
- `e(k_aux)` number of auxiliary parameters
- `e(k_eq)` number of equations in `e(b)`
- `e(k_eq_model)` number of equations in overall model test
- `e(df_m)` model degrees of freedom
- `e(ll)` log likelihood
- `e(ll_0)` log likelihood, constant-only model
- `e(N_clust)` number of clusters
- `e(chi2)` \( \chi^2 \)
- `e(p)` \( p \)-value for model \( \chi^2 \) test
- `e(sigma)` sigma
- `e(se_sigma)` standard error of sigma
- `e(rank)` rank of `e(V)`
- `e(rank0)` rank of `e(V)` for constant-only model
- `e(ic)` number of iterations
- `e(rc)` return code
- `e(converged)` 1 if converged, 0 otherwise

Macros

- `e(cmd)` `intreg`
- `e(cmdline)` command as typed
- `e(depvar)` names of dependent variables
- `e(wtype)` weight type
- `e(wexp)` weight expression
- `e(title)` title in estimation output
- `e(clustvar)` name of cluster variable
- `e(offset)` linear offset variable
- `e(chi2type)` Wald or LR; type of model \( \chi^2 \) test
- `e(vce)` `vce()`
- `e(vcetype)` title used to label Std. Err.
- `e(het)` heteroskedasticity, if `het()` specified
- `e(ml_score)` program used to implement scores
- `e(opt)` type of optimization
- `e(which)` max or min; whether optimizer is to perform maximization or minimization
- `e(ml_method)` type of `ml` method
- `e(user)` name of likelihood-evaluator program
- `e(technique)` maximization technique
- `e(properties)` `b V`
- `e(predict)` program used to implement `predict`
- `e(marginsok)` predictions allowed 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 (up to 20 iterations)
- `e(gradient)` gradient vector
- `e(V)` variance–covariance matrix of the estimators
- `e(V_modelbased)` model-based variance

Functions

- `e(sample)` marks estimation sample
Methods and formulas

The regression equation of interest is

\[ y_j = x_j \beta + \epsilon_j \]

where \( y_j \) is a continuous outcome for the \( j \)th observation—either observed or unobserved—with covariates \( x_j \) and corresponding coefficients \( \beta \). The model assumes that the error term is normally distributed; \( \epsilon \sim N(0, \sigma^2) \).

For observations \( j \in C \), we observe \( y_j \), that is, point data. Observations \( j \in I \) are intervals; we know only that the unobserved \( y_j \) is in the interval \( [y_{1j}, y_{2j}] \). For these observations, the likelihood contribution is \( \Pr(y_{1j} \leq Y_j \leq y_{2j}) \), where \( Y_j \) denotes the random variable representing the dependent variable in the model. Observations \( j \in L \) are left-censored; we know only that the unobserved \( y_j \) is less than or equal to \( y_{Lj} \), a censoring value that we do know. Similarly, observations \( j \in R \) are right-censored; we know only that the unobserved \( y_j \) is greater than or equal to \( y_{Rj} \). The likelihoods for these censored observations contain terms of the form \( \Pr(Y_j \leq y_{Lj}) \) for left-censored data and \( \Pr(Y_j \geq y_{Rj}) \) for right-censored data.

The log likelihood is

\[
\ln L = -\frac{1}{2} \sum_{j \in C} w_j \left\{ \frac{(y_j - x_j \beta)^2}{\sigma^2} + \log 2\pi \sigma^2 \right\} + \sum_{j \in L} w_j \log \Phi \left( \frac{y_{Lj} - x_j \beta}{\sigma} \right) + \sum_{j \in R} w_j \log \left\{ 1 - \Phi \left( \frac{y_{Rj} - x_j \beta}{\sigma} \right) \right\} + \sum_{j \in I} w_j \log \left\{ \Phi \left( \frac{y_{2j} - x_j \beta}{\sigma} \right) - \Phi \left( \frac{y_{1j} - x_j \beta}{\sigma} \right) \right\}
\]

where \( \Phi() \) is the cumulative standard normal distribution and \( w_j \) is the weight for the \( j \)th observation. If no weights are specified, \( w_j = 1 \). If \text{aweights} are specified, \( w_j = 1 \), and \( \sigma \) is replaced by \( \sigma / \sqrt{a_j} \) in the above, where \( a_j \) are the \text{aweights} normalized to sum to \( N \).

When the \text{het()} option is specified, \( \sigma \) is modeled as \( \ln(\sigma) = z_j^\gamma \), where \( z \) represents the variables in \text{het()} and \( \gamma \) is a vector of the estimated parameters to model the variance.

Note that the likelihood for \text{intreg} subsumes that of the \text{tobit} models; see \text{[R] tobit}.

Maximization is as described in \text{[R] Maximize}. \text{intreg} stores the estimated \( \sigma \) in \( e(b) \) in the log metric; therefore, if you want to provide an initial value for \( \sigma \) or to specify a constraint on it, ensure you do so on the log scale.

This command supports the Huber/White/sandwich estimator of the variance and its clustered version using \text{vce(robust)} and \text{vce(cluster clustvar)}, respectively. See \text{[P] _robust}, particularly \text{Maximum likelihood estimators} and \text{Methods and formulas}.

\text{intreg} also supports estimation with survey data. For details on VCEs with survey data, see \text{[SVY] Variance estimation}.
References


Also see

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

[R] *regress* — Linear regression

[R] *tobit* — Tobit regression

[BAYES] *bayes: intreg* — Bayesian interval regression

[ERM] *eintreg* — Extended interval regression

[FMM] *fmm: intreg* — Finite mixtures of interval regression models

[ME] *meintreg* — Multilevel mixed-effects interval regression

[ST] *stintreg* — Parametric models for interval-censored survival-time data

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

[XT] *xtintreg* — Random-effects interval-data regression models

[XT] *xttobit* — Random-effects tobit models

[U] 20 Estimation and postestimation commands