

# Interval Regression

Richard Williams, University of Notre Dame, <https://www3.nd.edu/~rwilliam/>  
Last revised September 29, 2024

**Overview.** From the Stata 17 Reference Manual:

`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.

Thus, with `intreg`, you have *two* dependent variables, representing the lower and upper bounds of the interval the respondent falls in.

`depvar1` and `depvar2` should have the following form:

Type of data		<code>depvar<sub>1</sub></code>	<code>depvar<sub>2</sub></code>
point data	$a = [a, a]$	$a$	$a$
interval data	$[a, b]$	$a$	$b$
left-censored data	$(-\infty, b]$	.	$b$
right-censored data	$[a, +\infty)$	$a$	.
missing		.	.

**Example 1.** As the Stata 17 manual notes, “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`.”

```
. webuse womenwage2, clear
(Wages of women, fictional data)
. des

Contains data from https://www.stata-press.com/data/r17/womenwage2.dta
Observations:      488      Wages of women, fictional data
Variables:         9        3 Jan 2021 13:00
```

Variable name	Storage type	Display format	Value label	Variable label
<code>wage1</code>	byte	%9.0g		Wage lower endpoint (\$1000s)
<code>wage2</code>	byte	%9.0g		Wage upper endpoint (\$1000s)
<code>age</code>	byte	%8.0g		Age in current year
<code>nev_mar</code>	byte	%8.0g		1 if never married
<code>rural</code>	byte	%8.0g		1 if not SMSA
<code>school</code>	byte	%8.0g		Current grade completed
<code>tenure</code>	float	%9.0g		Job tenure, in years
<code>wage</code>	byte	%9.0g		Wages in 1000s of dollars
<code>wagecat</code>	byte	%9.0g		Wage category (\$1000s)

```

. * Add value labels for wagecat
. label define wagecat 5 "$5,000 or less" 10 "$5,001 to $10,000" ///
> 15 "$10,001 to $15,000" 20 "$15,001 to $20,000" ///
> 25 "$20,001 to $25,000" 30 "$25,001 to $30,000" ///
> 40 "$30,001 to $40,000" 50 "$40,001 to $50,000" ///
> 51 "More than $50,000"
. label values wagecat wagecat
. fre wagecat

```

```
wagecat -- Wage category ($1000s)
```

		Freq.	Percent	Valid	Cum.
Valid	5 \$5,000 or less	14	2.87	2.87	2.87
	10 \$5,001 to \$10,000	83	17.01	17.01	19.88
	15 \$10,001 to \$15,000	158	32.38	32.38	52.25
	20 \$15,001 to \$20,000	107	21.93	21.93	74.18
	25 \$20,001 to \$25,000	57	11.68	11.68	85.86
	30 \$25,001 to \$30,000	30	6.15	6.15	92.01
	40 \$30,001 to \$40,000	19	3.89	3.89	95.90
	50 \$40,001 to \$50,000	14	2.87	2.87	98.77
	51 More than \$50,000	6	1.23	1.23	100.00
	Total	488	100.00	100.00	

Two recode commands can get you the upper and lower bounds of the intervals.

```

. * wage1 and wage2 are already in the dataset but we will
. * re-compute them to show how it is done.
. rename (wage1 wage2) (xwage1 xwage2)
. recode wagecat (5=.) (10=5) (15=10) (20=15) (25=20) ///
> (30=25) (40=30) (50=40) (51=50), gen(wage1)
(488 differences between wagecat and wage1)
. recode wagecat(51=.) , gen(wage2)
(6 differences between wagecat and wage2)
. label variable wage1 "Wage lower endpoint ($1000s)"
. label variable wage2 "Wage upper endpoint ($1000s)"

. * List a few cases
. set seed 123456
. bysort wagecat: gen firstcase = 1 if _n==1
(479 missing values generated)
. list wagecat wage1 wage2 if firstcase == 1

```

```

+-----+
|          wagecat   wage1   wage2 |
+-----+-----+-----+
  1. |      $5,000 or less      .     5 |
 15. |    $5,001 to $10,000      5    10 |
 98. |   $10,001 to $15,000     10   15 |
256. |   $15,001 to $20,000     15   20 |
363. |   $20,001 to $25,000     20   25 |
+-----+-----+-----+
420. |   $25,001 to $30,000     25   30 |
450. |   $30,001 to $40,000     30   40 |
469. |   $40,001 to $50,000     40   50 |
483. |   More than $50,000      50   . |
+-----+-----+-----+

```

```
. * Run intreg
. intreg wage1 wage2 c.age c.age#c.age i.nev_mar i.rural school tenure, nolog
```

```
Interval regression                                Number of obs    =    488
                                                Uncensored      =     0
                                                Left-censored   =    14
                                                Right-censored  =     6
                                                Interval-cens.  =   468

Log likelihood = -856.33293                      LR chi2(6)       = 221.61
                                                Prob > chi2     = 0.0000
```

	Coefficient	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	7.81265

We interpret results pretty much the same way we interpret results in an OLS regression. For example, those in rural areas make about \$3,000 a year less on average than do those in urban areas. Each year of schooling increases income by about \$1,334. Never-married people tend to make slightly less than ever-married people but the effect is not statistically significant.

*Using margins with intreg.* Given that `intreg` output looks much like the output from OLS regression, it is not surprising that `margins` produces similar looking output as it does for OLS.

```
. * AMEs
. margins, dydx(*)
```

```
Average marginal effects                                Number of obs = 488
Model VCE: OIM
```

```
Expression: Linear prediction, predict()
dy/dx wrt: age 1.nev_mar 1.rural school tenure
```

	Delta-method		z	P> z	[95% conf. interval]	
	dy/dx	std. err.				
age	.0294002	.0623938	0.47	0.637	-.0928894	.1516898
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

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

Note that the AMEs for 4 of the 5 variables are identical to the estimated coefficients. Age is different because the model actually includes age and age<sup>2</sup> and the AME reflects this. This is the same thing that happens with an OLS regression. Both the coefficients and the AMEs reflect linear effects of the independent variables on the dependent variable; whereas with commands like logit the independent variables have nonlinear effects on the probability of the event occurring.

See `help intreg_postestimation` for descriptions of other post-estimation commands and options after running `intreg`.

**Example 2.** Here is a hypothetical example using `intreg`. `y` is a continuous var that ranges from about -70 to 88. It is normally distributed. `ycat` is a collapsed, ordinal version of `y`. `y1` and `y2` are the upper and lower bounds of the `y` intervals.

```
. use "https://www3.nd.edu/~rwilliam/xsoc73994/statafiles/intreg.dta", clear
(Hypothetical data for intreg example)
```

```
. des
```

```
Contains data from D:\Soc73994\Statafiles\intreg.dta
obs:          1,000              Hypothetical data for intreg
                                example
                                6 Nov 2006 07:57
vars:          7
size:         32,000 (99.9% of memory free)
```

```
-----
```

variable name	storage type	display format	value label	variable label
y	float	%9.0g		Continuous Y, ranges from -70.4 to 88.06
ycat	float	%10.0g	ycat	Y collapsed into 5 intervals
y1	float	%9.0g		Lower bound of Y interval
y2	float	%9.0g		Upper bound of Y interval
x1	float	%9.0g		
x2	float	%9.0g		
x3	float	%9.0g		

```
-----
```

```
. sum y
```

```
-----
```

Variable	Obs	Mean	Std. Dev.	Min	Max
y	1000	14.01144	25.05774	-70.36776	88.0509

```
-----
```

```
. fre ycat
```

```
ycat -- Y collapsed into 5 intervals
```

```
-----
```

		Freq.	Percent	Valid	Cum.
Valid	1 LE 0	287	28.70	28.70	28.70
	2 0 to 15	224	22.40	22.40	51.10
	3 15 to 30	203	20.30	20.30	71.40
	4 30 to 45	183	18.30	18.30	89.70
	5 45 or more	103	10.30	10.30	100.00
	Total	1000	100.00	100.00	

```
-----
```

```
. * intreg with collapsed Y
. intreg y1 y2 x1 x2 x3, nolog
```

```
Interval regression                                Number of obs    = 1,000
                                                Uncensored      = 0
                                                Left-censored   = 287
                                                Right-censored  = 103
                                                Interval-cens.  = 610

Log likelihood = -1372.3949                    LR chi2(3)       = 386.33
                                                Prob > chi2     = 0.0000
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
x1	1.221547	.2544077	4.80	0.000	.7229169	1.720177
x2	.8989353	.0799428	11.24	0.000	.7422503	1.05562
x3	.9384835	.2191945	4.28	0.000	.5088702	1.368097
_cons	.0771196	1.451354	0.05	0.958	-2.767483	2.921722
/lnsigma	3.003777	.0320312	93.78	0.000	2.940997	3.066557
sigma	<b>20.16155</b>	.6457982			18.93472	21.46787

```
Observation summary:    287 left-censored observations
                        0 uncensored observations
                        103 right-censored observations
                        610 interval observations
```

```
. * OLS regression with original Y
. reg y x1 x2 x3
```

Source	SS	df	MS	Number of obs = 1000		
Model	227500.386	3	75833.4619	F( 3, 996)	=	188.94
Residual	399761.928	996	401.367397	Prob > F	=	0.0000
Total	627262.313	999	627.890204	R-squared	=	0.3627
				Adj R-squared	=	0.3608
				Root MSE	=	<b>20.034</b>

  

y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
x1	1.120216	.2308738	4.85	0.000	.6671616	1.573271
x2	.9312722	.0706904	13.17	0.000	.792553	1.069991
x3	.8474134	.1983744	4.27	0.000	.4581337	1.236693
_cons	.196622	1.245274	0.16	0.875	-2.247039	2.640284

Several things to note about the above:

- The nice thing about `intreg`, as opposed to other ordinal methods, is that you interpret its parameters the same way you do the parameters from an OLS regression. The sigma that `intreg` reports is equivalent to the root mean square error (i.e. the standard error of the residuals) from an OLS regression
- In this particular example, `intreg` does remarkably well. Its coefficients, standard errors, etc. are very similar to those produced by OLS regression on the un-collapsed `y` variable.

- I caution, however, that the example is “rigged” in `intreg`’s favor, in that the assumptions it makes about normality are true in the constructed data set. You can’t always count on it working this well. As the Stata manual notes, `intreg` assumes normality.

*Assessing how well `intreg` works in practice.* Of course, in real situations, we don’t know what the true value of Y is. If we did, we wouldn’t be using `intreg`. To address this problem, the Stata manual recommends estimating an `oprobit` model using `yecat` as the dependent variable and the same independent variables:

```
. * oprobit with collapsed Y
. oprobit ycat x1 x2 x3, nolog
```

```
Ordered probit regression                               Number of obs   =       1000
                                                         LR chi2(3)      =       386.49
                                                         Prob > chi2     =       0.0000
Log likelihood = -1368.7378                             Pseudo R2       =       0.1237
```

ycat	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
x1	.0604916	.0126526	4.78	0.000	.035693	.0852902
x2	.0445961	.004006	11.13	0.000	.0367445	.0524476
x3	.0466968	.0108907	4.29	0.000	.0253514	.0680421
/cut1	.0091044	.0732018			-.1343684	.1525773
/cut2	.7462179	.0751763			.5988751	.8935608
/cut3	1.415098	.0809962			1.256348	1.573848
/cut4	2.285878	.0952678			2.099156	2.472599

*The key is to compare the log-likelihoods of the `intreg` and `oprobit` models.* In this case, the log-likelihoods for `intreg` (-1372.3949) and `oprobit` (-1368.7378) are almost identical, meaning both models fit the data about equally well. The z values for the models are also about the same. (NOTE: You should compare the log-likelihoods rather than the model chi-squares when comparing `intreg` and `oprobit`.) Since both models fit equally well, you may want to use `intreg` because the coefficients from it are so much easier to interpret.

If, on the other hand, `oprobit` fits much better, the Stata manual suggests you might want to modify the `intreg` model (e.g. take logs of the interval points) or use `oprobit` or `ologit` or some other ordinal method instead. The Stata Reference Manual entry for `intreg` illustrates how to do this with Example 1, and this handout’s Appendix elaborates even further.

**Note:** The Stata 17 manual warns that the `oprobit/intreg` comparison is not always appropriate. “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.” In other words, if one of the intervals consisted of a single point, e.g. `devar1 = devar2 = 25`, you couldn’t use `oprobit` to test how well `intreg` was working.

## Appendix: Example 1 Revisited

The hypothetical data in Example 1 also includes the “real” value for age, so we can assess the `intreg` model the same way we did in Example 2. We run the `intreg` model, the corresponding `oprobit` model, and then the OLS regress model using real wage.

```
. *** Example 1 revisited ***
. webuse womenwage2, clear
(Wages of women, fictional data)
. intreg wage1 wage2 c.age c.age#c.age i.nev_mar i.rural school tenure, nolog
```

```
Interval regression                                Number of obs   =   488
                                                Uncensored     =    0
                                                Left-censored  =   14
                                                Right-censored =    6
                                                Interval-cens. =  468

                                                LR chi2(6)    =  221.61
                                                Prob > chi2   =  0.0000

Log likelihood = -856.33293
```

	Coefficient	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	<b>7.299626</b>	.2529634			6.82029	7.81265

```
. oprobit wagecat c.age c.age#c.age i.nev_mar i.rural school tenure, nolog
```

```
Ordered probit regression                                Number of obs   =   488
                                                LR chi2(6)    =  235.68
                                                Prob > chi2   =  0.0000
                                                Pseudo R2    =  0.1337

Log likelihood = -763.31049
```

wagecat	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
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	.127977
/cut1	2.650637	.8957245			.8950495	4.406225
/cut2	3.941018	.8979167			2.181134	5.700903
/cut3	5.085205	.9056582			3.310148	6.860263
/cut4	5.875534	.9120933			4.087864	7.663204
/cut5	6.468723	.918117			4.669247	8.268199
/cut6	6.922726	.9215455			5.11653	8.728922
/cut7	7.34471	.9237628			5.534168	9.155252
/cut8	7.963441	.9338881			6.133054	9.793828

In the above, `oprobit` fits much better than `intreg`, i.e. it has a much smaller log likelihood. Further, if we run the OLS regression with “real” wage, we see that the `intreg` and OLS estimates differ by fairly large amounts.

```
. reg wage c.age#c.age i.nev_mar i.rural school tenure
```

Source	SS	df	MS	Number of obs	=	488
-----				F(6, 481)	=	44.81
Model	17182.6184	6	2863.76974	Prob > F	=	0.0000
Residual	30741.3631	481	63.9113578	R-squared	=	0.3585
-----				Adj R-squared	=	0.3505
Total	47923.9816	487	98.406533	Root MSE	=	<b>7.9945</b>

wage	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
-----						
age	.5078072	.4708266	1.08	0.281	-.4173239	1.432938
c.age#c.age	-.0083304	.0077619	-1.07	0.284	-.0235819	.006921
1.nev_mar	-.1652674	.864845	-0.19	0.849	-1.864608	1.534074
1.rural	-2.915707	.8283239	-3.52	0.000	-4.543288	-1.288127
school	1.336653	.1444367	9.25	0.000	1.052848	1.620458
tenure	.8993539	.1110741	8.10	0.000	.6811034	1.117604
_cons	-8.409316	6.755676	-1.24	0.214	-21.6836	4.864966
-----						

Income is generally not normally distributed, which is a requirement for the use of `intreg`. The Stata Manual notes that “Normality is more closely approximated if we model the log of wages.” So, we will compute the logs of `wage1`, `wage2`, and `wage`, and see how well `intreg` works then. (There is no need to compute the log of `wagecat`, since the only thing that matters to `oprobit` is the ordering of categories, not their specific values.)

```
. * intreg doesn't work that well, so lets try log(wages) instead
. gen logwage1 = log(wage1)
(14 missing values generated)
. gen logwage2 = log(wage2)
(6 missing values generated)
. gen logwage = log(wage)
```



```
. intreg logwage1 logwage2 c.age c.age#c.age i.nev_mar i.rural school tenure, nolog
```

```
Interval regression                               Number of obs   =   488
                                                Uncensored     =     0
                                                Left-censored  =    14
                                                Right-censored =     6
                                                Interval-cens. =   468

                                                LR chi2(6)     =  231.40
Log likelihood = -773.36563                    Prob > chi2    =  0.0000
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
age	.0645589	.0249954	2.58	0.010	.0155689	.1135489
c.age#c.age	-.0010812	.0004115	-2.63	0.009	-.0018878	-.0002746
1.nev_mar	-.0058151	.0454867	-0.13	0.898	-.0949674	.0833371
1.rural	-.2098361	.0439454	-4.77	0.000	-.2959675	-.1237047
school	.0804832	.0076783	10.48	0.000	.0654341	.0955323
tenure	.0397144	.0058001	6.85	0.000	.0283464	.0510825
_cons	.7084023	.3593193	1.97	0.049	.0041495	1.412655
/lnsigma	-.906989	.0356265	-25.46	0.000	-.9768157	-.8371623
sigma	<b>.4037381</b>	.0143838			.3765081	.4329373

```
. oprobit wagecat c.age c.age#c.age i.nev_mar i.rural school tenure, nolog
```

```
Ordered probit regression                               Number of obs   =   488
                                                LR chi2(6)     =  235.68
                                                Prob > chi2    =  0.0000
Log likelihood = -763.31049                    Pseudo R2      =  0.1337
```

wagecat	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
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	.127977
/cut1	2.650637	.8957245			.8950495	4.406225
/cut2	3.941018	.8979167			2.181134	5.700903
/cut3	5.085205	.9056582			3.310148	6.860263
/cut4	5.875534	.9120933			4.087864	7.663204
/cut5	6.468723	.918117			4.669247	8.268199
/cut6	6.922726	.9215455			5.11653	8.728922
/cut7	7.34471	.9237628			5.534168	9.155252
/cut8	7.963441	.9338881			6.133054	9.793828

Now the fit of intreg and oprobit is almost identical. Further, the Z values for the coefficients are very similar.

Using OLS with the log of “real” wage,

```
. reg logwage c.age c.age#c.age i.nev_mar i.rural school tenure
```

Source	SS	df	MS	Number of obs	=	488
Model	48.3502704	6	8.0583784	F(6, 481)	=	51.73
Residual	74.9286926	481	.155776908	Prob > F	=	0.0000
				R-squared	=	0.3922
				Adj R-squared	=	0.3846
Total	123.278963	487	.253139554	Root MSE	=	<b>.39469</b>

  

logwage	Coefficient	Std. err.	t	P> t	[95% conf. interval]
age	.057249	.0232447	2.46	0.014	.0115753 .1029226
c.age#c.age	-.0009289	.0003832	-2.42	0.016	-.0016818 -.0001759
1.nev_mar	.0158159	.0426973	0.37	0.711	-.0680804 .0997123
1.rural	-.1902597	.0408943	-4.65	0.000	-.2706133 -.1099062
school	.0741954	.0071308	10.40	0.000	.060184 .0882068
tenure	.0399751	.0054837	7.29	0.000	.0292001 .0507502
_cons	.8962183	.3335272	2.69	0.007	.2408679 1.551569

We see that the coefficients and Z values for `intreg` and `regress` are very similar.

**Summary.** If you want to evaluate whether it is ok to use `intreg`,

- Run both `intreg` and the corresponding `oprobit` model. If the model fits (i.e. the Log Likelihoods) and coefficient Z values are similar, then you may want to use `intreg` because its coefficients can be much easier to interpret.
- If the fit of `oprobit` is much better than the fit of `intreg`, consider whether there is some transformation of the dependent variable that would work better. `intreg` assumes normality, and the log of income is more likely to be normally distributed than income is.
- If, after trying transformations of the dependent variable, `oprobit` still fits much better than `intreg`, then you probably don't want to use `intreg`. Use something like `oprobit` or `ologit` instead.