

Supplemental Notes on the intreg Command

From the Stata 11 manual, p. 712:

Hence, `intreg` can fit models for data where each observation represents interval data, left-censored data, right-censored data, or point data. Regardless of the type of observation, the data should be stored in the dataset as interval data; that is, two dependent variables, $depvar_1$ and $depvar_2$, 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, `.'`, or an extended missing value, `._a, .b, ..., .z`, in $depvar_1$. If the data are right-censored, the upper endpoint is $+\infty$ and is represented by a missing value, `.'` (or an extended missing value), in $depvar_2$. Point data are represented by the two endpoints being equal.

type of data		$depvar_1$	$depvar_2$
point data	$a = [a, a]$	a	a
interval data	$[a, b]$	a	b
left-censored data	$(-\infty, b]$	<code>.</code>	b
right-censored data	$[a, +\infty)$	a	<code>.</code>

Truly missing values of the dependent variable must be represented by missing values in both $depvar_1$ and $depvar_2$.

Example 1. As the Stata 11 manual notes, “Women were asked via a questionnaire to indicate a category for their yearly income from employment. The categories were less than 5,000, 5,001–10,000, ... , 25,001–30,000, 30,001–40,000, 40,001–50,000, and more than 50,000. The wage categories are stored in the `wagecat` variable... A value of 5 for `wagecat` represents the category less than 5,000, a value of 10 represents 5,001–10,000, ... , and a value of 51 represents greater than 50,000. To use `intreg`, we must create two variables, `wage1` and `wage2`, containing the lower and upper endpoints of the wage categories.”

I think the Stata documentation makes the construction of the data set much more complicated than is necessary. Two `recode` commands can get you the upper and lower bounds of the intervals. Here is a simpler solution.

```
. webuse womenwage, clear
(Wages of women)
```

```
. tab1 wagecat
```

```
-> tabulation of wagecat
```

Wage category (\$1000s)	Freq.	Percent	Cum.
5	14	2.87	2.87
10	83	17.01	19.88
15	158	32.38	52.25
20	107	21.93	74.18
25	57	11.68	85.86
30	30	6.15	92.01
40	19	3.89	95.90
50	14	2.87	98.77
51	6	1.23	100.00
Total	488	100.00	

```
. 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)
```

```
. sort age, stable
```

```
. list wage1 wage2 in 1/10
```

	wage1	wage2
1.	.	5
2.	5	10
3.	5	10
4.	10	15
5.	.	5
6.	.	5
7.	.	5
8.	5	10
9.	5	10
10.	5	10

```
. intreg wage1 wage2 age age2 nev_mar 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                                Number of obs =      488
Log likelihood = -856.33293                       LR chi2(6)         =    221.61
                                                    Prob > chi2        =    0.0000
```

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
age	.7914438	.4433604	1.79	0.074	-.0775265	1.660414
age2	-.0132624	.0073028	-1.82	0.069	-.0275757	.0010509
nev_mar	-.2075022	.8119581	-0.26	0.798	-1.798911	1.383906
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

```
Observation summary:      14 left-censored observations
                          0 uncensored observations
                          6 right-censored observations
                          468 interval observations
```

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 "http://www.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
vars:      7      6 Nov 2006 07:57
size:     32,000 (99.9% of memory free)
```

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

```
Sorted by:
```

```
. sum y
```

Variable	Obs	Mean	Std. Dev.	Min	Max
<code>y</code>	1000	14.01144	25.05774	-70.36776	88.0509

```
. tabl ycat
```

```
-> tabulation of ycat
```

Y collapsed into 5 intervals	Freq.	Percent	Cum.
LE 0	287	28.70	28.70
0 to 15	224	22.40	51.10
15 to 30	203	20.30	71.40
30 to 45	183	18.30	89.70
45 or more	103	10.30	100.00
Total	1,000	100.00	

```
. * intreg with collapsed Y
```

```
. intreg y1 y2 x1 x2 x3
```

```
Fitting constant-only model:
```

```
Iteration 0: log likelihood = -1688.3436  
Iteration 1: log likelihood = -1574.6026  
Iteration 2: log likelihood = -1565.5637  
Iteration 3: log likelihood = -1565.5603  
Iteration 4: log likelihood = -1565.5603
```

```
Fitting full model:
```

```
Iteration 0: log likelihood = -1508.2373  
Iteration 1: log likelihood = -1379.0543  
Iteration 2: log likelihood = -1372.4038  
Iteration 3: log likelihood = -1372.3949  
Iteration 4: log likelihood = -1372.3949
```

```
Interval regression                                Number of obs =      1000  
                                                    LR chi2(3)         =      386.33  
Log likelihood = -1372.3949                      Prob > chi2        =      0.0000
```

	Coef.	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	20.16155	.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			
Model	227500.386	3	75833.4619	Number of obs =	1000	
Residual	399761.928	996	401.367397	F(3, 996) =	188.94	
				Prob > F =	0.0000	
				R-squared =	0.3627	
				Adj R-squared =	0.3608	
Total	627262.313	999	627.890204	Root MSE =	20.034	

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

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

```
Iteration 0: log likelihood = -1561.9813
Iteration 1: log likelihood = -1370.1889
Iteration 2: log likelihood = -1368.7383
Iteration 3: log likelihood = -1368.7378
```

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

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.
- Also, `intreg` produces almost the exact same log-likelihood as does `oprobit`, and also the same z values. (NOTE: You should compare the log-likelihoods rather than the model chi-squares when comparing `intreg` and `oprobit`.) But, the coefficients from `intreg` are much easier to interpret.
 - As the Stata manual points out, if `oprobit` fit much better, 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.
- 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.