Title

stata.com

nestreg — Nested model statistics

Description

`nestreg` fits nested models by sequentially adding blocks of variables and then reports comparison tests between the nested models.

Quick start

Fit nested (hierarchical) models sequentially, including covariates `x1` and `x2` first and then adding `x3` and `x4`

```
nestreg: regress y (x1 x2) (x3 x4)
```

Also fit third model including `d1`, `d2`, and `d3`

```
nestreg: regress y (x1 x2) (x3 x4) (d1 d2 d3)
```

Report table of likelihood-ratio tests instead of Wald tests comparing models

```
nestreg, lrtable: regress y (x1 x2) (x3 x4) (d1 d2 d3)
```

Fit nested models and adjust for complex survey design using `svyset` data

```
nestreg: svy: regress y (x1 x2) (x3 x4) (d1 d2 d3)
```

Note: In the above examples, `regress` could be replaced with any estimation command allowing the `nestreg` prefix.

Menu

Statistics > Other > Nested model statistics

Syntax

Standard estimation command syntax

```
nestreg [ , options ]: command_name depvar (varlist) [ (varlist) ... ]
    [ if ] [ in ] [ weight ] [ , command_options ]
```

Survey estimation command syntax

```
nestreg [ , options ]: svy [ vcetype ] [ , svy_options ]: command_name depvar
    (varlist) [ (varlist) ... ] [ if ] [ in ] [ , command_options ]
```

options	Description
Reporting	
`waldtable`	report Wald test results; the default
`lrtable`	report likelihood-ratio test results
`quietly`	suppress any output from *command_name*
`store(`*stub*`)`	store nested estimation results in `_est_`*stub*`#`

`by` is allowed; see [U] **11.1.10 Prefix commands**.

Weights are allowed if *command_name* allows them; see [U] **11.1.6 weight**.

A *varlist* in parentheses indicates that this list of variables is to be considered as a block. Each variable in a *varlist* not bound in parentheses will be treated as its own block.

All postestimation commands behave as they would after *command_name* without the `nestreg` prefix; see the postestimation manual entry for *command_name*.

Options

Reporting

`waldtable` specifies that the table of Wald test results be reported. `waldtable` is the default.

`lrtable` specifies that the table of likelihood-ratio tests be reported. This option is not allowed if `pweights`, the `vce(robust)` option, or the `vce(cluster` *clustvar*`)` option is specified. `lrtable` is also not allowed with the `svy` prefix.

`quietly` suppresses the display of any output from *command_name*.

`store(`*stub*`)` specifies that each model fit by `nestreg` be stored under the name `_est_`*stub*`#`, where `#` is the nesting order from first to last.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Estimation commands

`nestreg` removes collinear predictors and observations with missing values from the estimation sample before calling *command_name*.

The following Stata commands are supported by `nestreg`:

betareg, clogit, cloglog, glm, intreg, logistic, logit, nbreg, ologit, oprobit, poisson, probit, qreg, regress, scobit, stcox, stcrreg, stintreg, streg, and tobit

You do not supply a *depvar* for `stcox`, `stintreg`, `stcrreg`, or `streg`; otherwise, *depvar* is required. You must supply two *depvar*s for `intreg`.

Wald tests

Use `nestreg` to test the significance of blocks of predictors, building the regression model one block at a time. Using the data from example 1 of [R] **test**, we wish to test the significance of the following predictors of birth rate: `medage`, `medagesq`, and `region` (already partitioned into four indicator variables: `reg1`, `reg2`, `reg3`, and `reg4`).

```
. use http://www.stata-press.com/data/r15/census4
(birth rate, median age)
. nestreg: regress brate (medage) (medagesq) (reg2-reg4)
Block  1: medage
```

Source	SS	df	MS	Number of obs	=	50
				F(1, 48)	=	164.72
Model	32675.1044	1	32675.1044	Prob > F	=	0.0000
Residual	9521.71561	48	198.369075	R-squared	=	0.7743
				Adj R-squared	=	0.7696
Total	42196.82	49	861.159592	Root MSE	=	14.084

brate	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
medage	-15.24893	1.188141	-12.83	0.000	-17.63785	-12.86002
_cons	618.3935	35.15416	17.59	0.000	547.7113	689.0756

```
Block  2: medagesq
```

Source	SS	df	MS	Number of obs	=	50
				F(2, 47)	=	158.75
Model	36755.8524	2	18377.9262	Prob > F	=	0.0000
Residual	5440.96755	47	115.765267	R-squared	=	0.8711
				Adj R-squared	=	0.8656
Total	42196.82	49	861.159592	Root MSE	=	10.759

brate	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
medage	-109.8925	15.96663	-6.88	0.000	-142.0132	-77.7718
medagesq	1.607332	.2707228	5.94	0.000	1.062708	2.151956
_cons	2007.071	235.4316	8.53	0.000	1533.444	2480.698

```
Block  3: reg2 reg3 reg4
```

Source	SS	df	MS	Number of obs	=	50
				F(5, 44)	=	100.63
Model	38803.419	5	7760.68381	Prob > F	=	0.0000
Residual	3393.40095	44	77.1227489	R-squared	=	0.9196
				Adj R-squared	=	0.9104
Total	42196.82	49	861.159592	Root MSE	=	8.782

brate	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
medage	-109.0957	13.52452	-8.07	0.000	-136.3526	-81.83886
medagesq	1.635208	.2290536	7.14	0.000	1.173581	2.096835
reg2	15.00284	4.252068	3.53	0.001	6.433365	23.57233
reg3	7.366435	3.953336	1.86	0.069	-.6009898	15.33386
reg4	21.39679	4.650602	4.60	0.000	12.02412	30.76946
_cons	1947.61	199.8405	9.75	0.000	1544.858	2350.362

Block	F	Block df	Residual df	Pr > F	R2	Change in R2
1	164.72	1	48	0.0000	0.7743	
2	35.25	1	47	0.0000	0.8711	0.0967
3	8.85	3	44	0.0001	0.9196	0.0485

This single call to nestreg ran regress three times, adding a block of predictors to the model for each run as in

```
. regress brate medage
      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(1, 48)      =  164.72
       Model |  32675.1044     1  32675.1044            Prob > F      =  0.0000
    Residual |  9521.71561    48  198.369075            R-squared     =  0.7743
-------------+------------------------------           Adj R-squared =  0.7696
       Total |   42196.82     49  861.159592            Root MSE      =  14.084

------------------------------------------------------------------------------
       brate |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      medage |  -15.24893   1.188141   -12.83   0.000    -17.63785   -12.86002
       _cons |   618.3935   35.15416    17.59   0.000     547.7113    689.0756
------------------------------------------------------------------------------

. regress brate medage medagesq
      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(2, 47)      =  158.75
       Model |  36755.8524     2  18377.9262            Prob > F      =  0.0000
    Residual |  5440.96755    47  115.765267            R-squared     =  0.8711
-------------+------------------------------           Adj R-squared =  0.8656
       Total |   42196.82     49  861.159592            Root MSE      =  10.759

------------------------------------------------------------------------------
       brate |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      medage |  -109.8925   15.96663    -6.88   0.000    -142.0132   -77.7718
    medagesq |   1.607332   .2707228     5.94   0.000     1.062708   2.151956
       _cons |   2007.071   235.4316     8.53   0.000     1533.444   2480.698
------------------------------------------------------------------------------

. regress brate medage medagesq reg2-reg4
      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(5, 44)      =  100.63
       Model |  38803.419     5  7760.68381            Prob > F      =  0.0000
    Residual |  3393.40095    44  77.1227489            R-squared     =  0.9196
-------------+------------------------------           Adj R-squared =  0.9104
       Total |   42196.82     49  861.159592            Root MSE      =   8.782

------------------------------------------------------------------------------
       brate |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      medage |  -109.0957   13.52452    -8.07   0.000    -136.3526   -81.83886
    medagesq |   1.635208   .2290536     7.14   0.000     1.173581   2.096835
        reg2 |   15.00284   4.252068     3.53   0.001     6.433365   23.57233
        reg3 |   7.366435   3.953336     1.86   0.069    -.6009898   15.33386
        reg4 |   21.39679   4.650602     4.60   0.000     12.02412   30.76946
       _cons |   1947.61    199.8405     9.75   0.000     1544.858    2350.362
------------------------------------------------------------------------------
```

nestreg collected the F statistic for the corresponding block of predictors and the model R^2 statistic from each model fit.

The F statistic for the first block, 164.72, is for a test of the joint significance of the first block of variables; it is simply the F statistic from the regression of brate on medage. The F statistic for the second block, 35.25, is for a test of the joint significance of the second block of variables in a regression of both the first and second blocks of variables. In our example, it is an F test of medagesq in the regression of brate on medage and medagesq. Similarly, the third block's F statistic of 8.85 corresponds to a joint test of reg2, reg3, and reg4 in the final regression.

Likelihood-ratio tests

The nestreg command provides a simple syntax for performing likelihood-ratio tests for nested model specifications; also see lrtest. Using the data from example 1 of [R] lrtest, we wish to jointly test the significance of the following predictors of low birthweight: age, lwt, ptl, and ht.

```
. use http://www.stata-press.com/data/r15/lbw
(Hosmer & Lemeshow data)
. xi: nestreg, lr: logistic low (i.race smoke ui) (age lwt ptl ht)
i.race            _Irace_1-3          (naturally coded; _Irace_1 omitted)
Block  1: _Irace_2 _Irace_3 smoke ui
Logistic regression                               Number of obs     =        189
                                                  LR chi2(4)        =      18.80
                                                  Prob > chi2       =     0.0009
Log likelihood = -107.93404                       Pseudo R2         =     0.0801
```

low	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_Irace_2	3.052746	1.498087	2.27	0.023	1.166747	7.987382
_Irace_3	2.922593	1.189229	2.64	0.008	1.316457	6.488285
smoke	2.945742	1.101838	2.89	0.004	1.415167	6.131715
ui	2.419131	1.047359	2.04	0.041	1.035459	5.651788
_cons	.1402209	.0512295	-5.38	0.000	.0685216	.2869447

```
Note: _cons estimates baseline odds.
Block  2: age lwt ptl ht
Logistic regression                               Number of obs     =        189
                                                  LR chi2(8)        =      33.22
                                                  Prob > chi2       =     0.0001
Log likelihood =   -100.724                       Pseudo R2         =     0.1416
```

low	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_Irace_2	3.534767	1.860737	2.40	0.016	1.259736	9.918406
_Irace_3	2.368079	1.039949	1.96	0.050	1.001356	5.600207
smoke	2.517698	1.00916	2.30	0.021	1.147676	5.523162
ui	2.1351	.9808153	1.65	0.099	.8677528	5.2534
age	.9732636	.0354759	-0.74	0.457	.9061578	1.045339
lwt	.9849634	.0068217	-2.19	0.029	.9716834	.9984249
ptl	1.719161	.5952579	1.56	0.118	.8721455	3.388787
ht	6.249602	4.322408	2.65	0.008	1.611152	24.24199
_cons	1.586014	1.910496	0.38	0.702	.1496092	16.8134

```
Note: _cons estimates baseline odds.
```

Block	LL	LR	df	Pr > LR	AIC	BIC
1	-107.934	18.80	4	0.0009	225.8681	242.0768
2	-100.724	14.42	4	0.0061	219.448	248.6237

The estimation results from the full model are left in e(), so we can later use estat and other postestimation commands.

```
. estat gof
Logistic model for low, goodness-of-fit test
       number of observations =       189
 number of covariate patterns =       182
           Pearson chi2(173) =       179.24
                 Prob > chi2 =         0.3567
```

Programming for nestreg

If you want your community-contributed command (*command_name*) to work with `nestreg`, it must follow standard Stata syntax and allow the `if` qualifier. Furthermore, *command_name* must have `sw` or `swml` as a program property; see [P] **program properties**. If *command_name* has `swml` as a property, *command_name* must store the log-likelihood value in `e(ll)` and the model degrees of freedom in `e(df_m)`.

Stored results

`nestreg` stores the following in `r()`:

Matrices

`r(wald)`	matrix corresponding to the Wald table
`r(lr)`	matrix corresponding to the likelihood-ratio table

Acknowledgment

We thank Paul H. Bern of Syracuse University for developing the hierarchical regression command that inspired `nestreg`.

References

Acock, A. C. 2018. *A Gentle Introduction to Stata*. 6th ed. College Station, TX: Stata Press.

Lindsey, C., and S. J. Sheather. 2015. Best subsets variable selection in nonnormal regression models. *Stata Journal* 15: 1046–1059.

Also see

[P] **program properties** — Properties of user-defined programs