stepwise — Stepwise estimation

Description

`stepwise` performs stepwise estimation. Typing

`. stepwise, pr(#):` *command*

performs backward-selection estimation for *command*. The stepwise selection method is determined by the following option combinations:

options	Description
`pr(#)`	backward selection
`pr(#) hierarchical`	backward hierarchical selection
`pr(#) pe(#)`	backward stepwise
`pe(#)`	forward selection
`pe(#) hierarchical`	forward hierarchical selection
`pr(#) pe(#) forward`	forward stepwise

command defines the estimation command to be executed. The following Stata commands are supported by `stepwise`:

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

`stepwise` expects *command* to have the following form:

command_name [*depvar*] *term* [*term* ...] [*if*] [*in*] [*weight*] [, *command_options*]

where *term* is either *varname* or (*varlist*) (a *varlist* in parentheses indicates that this group of variables is to be included or excluded together). *depvar* is not present when *command_name* is `stcox`, `stcrreg`, or `streg`; otherwise, *depvar* is assumed to be present. For `intreg`, *depvar* is actually two dependent variable names ($depvar_1$ and $depvar_2$).

`sw` is a synonym for `stepwise`.

Quick start

Backward selection, removing terms with $p \geq 0.2$

```
stepwise, pr(.2): regress y x1 x2 x3 x4
```

As above, but force `x1` to be included in model

```
stepwise, pr(.2) lockterm1: regress y x1 x2 x3 x4
```

Add `d1`, `d2`, and `d3` and consider as a group for inclusion in model

```
stepwise, pr(.2): regress y x1 x2 x3 x4 (d1 d2 d3)
```

Force `d1`, `d2`, and `d3` to be included in model

```
stepwise, pr(.2) lockterm1: regress y (d1 d2 d3) x1 x2 x3 x4
```

Forward selection, adding terms with $p < 0.1$

```
stepwise, pe(.1): regress y x1 x2 x3 x4
```

Backward stepwise selection, removing terms with $p \geq 0.2$ and adding those with $p < 0.1$

```
stepwise, pr(.2) pe(.1): regress y x1 x2 x3 x4
```

Forward stepwise selection, adding terms with $p < 0.1$ and removing those with $p \geq 0.2$

```
stepwise, pr(.2) pe(.1) forward: regress y x1 x2 x3 x4
```

Backward hierarchical selection

```
stepwise, pr(.2) hierarchical: regress y x1 x2 x3 x4
```

Forward hierarchical selection

```
stepwise, pe(.1) hierarchical: regress y x1 x2 x3 x4
```

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

Menu

Statistics > Other > Stepwise estimation

Syntax

`stepwise` [, *options*] : *command*

options	Description
Model	
* pr(#)	significance level for removal from the model
* pe(#)	significance level for addition to the model
Model2	
forward	perform forward-stepwise selection
hierarchical	perform hierarchical selection
lockterm1	keep the first term
lr	perform likelihood-ratio test instead of Wald test
Reporting	
display_options	control columns and column formats and line width

* At least one of `pr(#)` or `pe(#)` must be specified.

`by` and `xi` are allowed; see **[U] 11.1.10 Prefix commands**.

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

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

Options

Model

`pr(#)` specifies the significance level for removal from the model; terms with $p \geq$ `pr()` are eligible for removal.

`pe(#)` specifies the significance level for addition to the model; terms with $p <$ `pe()` are eligible for addition.

Model 2

`forward` specifies the forward-stepwise method and may be specified only when both `pr()` and `pe()` are also specified. Specifying both `pr()` and `pe()` without `forward` results in backward-stepwise selection. Specifying only `pr()` results in backward selection, and specifying only `pe()` results in forward selection.

`hierarchical` specifies hierarchical selection.

`lockterm1` specifies that the first term be included in the model and not be subjected to the selection criteria.

`lr` specifies that the test of term significance be the likelihood-ratio test. The default is the less computationally expensive Wald test; that is, the test is based on the estimated variance–covariance matrix of the estimators.

Reporting

display_options: `noci`, `nopvalues`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see **[R] estimation options**.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Introduction

Typing

```
. stepwise, pr(.10): regress y1 x1 x2 d1 d2 d3 x4 x5
```

performs a backward-selection search for the regression model `y1` on `x1`, `x2`, `d1`, `d2`, `d3`, `x4`, and `x5`. In this search, each explanatory variable is said to be a term. Typing

```
. stepwise, pr(.10): regress y1 x1 x2 (d1 d2 d3) (x4 x5)
```

performs a similar backward-selection search, but the variables `d1`, `d2`, and `d3` are treated as one term, as are `x4` and `x5`. That is, `d1`, `d2`, and `d3` may or may not appear in the final model, but they appear or do not appear together.

▷ Example 1

Using the automobile dataset, we fit a backward-selection model of `mpg`:

```
. use http://www.stata-press.com/data/r14/auto
. generate weight2 = weight*weight
. stepwise, pr(.2): regress mpg weight weight2 displ gear turn headroom foreign
> price
                      begin with full model
p = 0.7116 >= 0.2000  removing headroom
p = 0.6138 >= 0.2000  removing displacement
p = 0.3278 >= 0.2000  removing price

      Source |       SS           df       MS      Number of obs   =        74
-------------+----------------------------------   F(5, 68)        =     33.39
       Model |  1736.31455         5  347.262911   Prob > F        =    0.0000
    Residual |  707.144906        68  10.3991898   R-squared       =    0.7106
-------------+----------------------------------   Adj R-squared   =    0.6893
       Total |  2443.45946        73  33.4720474   Root MSE        =    3.2248

------------------------------------------------------------------------------
         mpg |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |  -.0158002   .0039169    -4.03   0.000    -.0236162   -.0079842
     weight2 |   1.77e-06   6.20e-07     2.86   0.006     5.37e-07    3.01e-06
     foreign |  -3.615107   1.260844    -2.87   0.006    -6.131082   -1.099131
  gear_ratio |   2.011674   1.468831     1.37   0.175    -.9193321     4.94268
        turn |  -.3087038   .1763099    -1.75   0.084    -.6605248    .0431172
       _cons |   59.02133     9.3903     6.29   0.000     40.28327    77.75938
------------------------------------------------------------------------------
```

This estimation treated each variable as its own term and thus considered each one separately. The engine displacement and gear ratio should really be considered together:

```
. stepwise, pr(.2): regress mpg weight weight2 (displ gear) turn headroom
> foreign price
                      begin with full model
p = 0.7116 >= 0.2000  removing headroom
p = 0.3944 >= 0.2000  removing displacement gear_ratio
p = 0.2798 >= 0.2000  removing price
      Source |       SS           df       MS      Number of obs   =        74
-------------+----------------------------------   F(4, 69)        =     40.76
       Model |  1716.80842         4  429.202105   Prob > F        =    0.0000
    Residual |  726.651041        69  10.5311745   R-squared       =    0.7026
-------------+----------------------------------   Adj R-squared   =    0.6854
       Total |  2443.45946        73  33.4720474   Root MSE        =    2.4452

------------------------------------------------------------------------------
         mpg |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |  -.0160341   .0039379    -4.07   0.000    -.0238901   -.0081782
     weight2 |   1.70e-06   6.21e-07     2.73   0.008     4.58e-07    2.94e-06
     foreign |  -2.758668   1.101772    -2.50   0.015    -4.956643   -.5606925
        turn |  -.2862724    .176658    -1.62   0.110    -.6386955    .0661508
       _cons |   65.39216   8.208778     7.97   0.000      49.0161    81.76823
------------------------------------------------------------------------------
```

◁

Search logic for a step

Before discussing the complete search logic, consider the logic for a step—the first step—in detail. The other steps follow the same logic. If you type

```
. stepwise, pr(.20): regress y1 x1 x2 (d1 d2 d3) (x4 x5)
```

the logic is

1. Fit the model y on x1 x2 d1 d2 d3 x4 x5.
2. Consider dropping x1.
3. Consider dropping x2.
4. Consider dropping d1 d2 d3.
5. Consider dropping x4 x5.
6. Find the term above that is least significant. If its significance level is ≥ 0.20, remove that term.

If you type

```
. stepwise, pr(.20) hierarchical: regress y1 x1 x2 (d1 d2 d3) (x4 x5)
```

the logic would be different because the hierarchical option states that the terms are ordered. The initial logic would become

1. Fit the model y on x1 x2 d1 d2 d3 x4 x5.
2. Consider dropping x4 x5—the last term.
3. If the significance of this last term is ≥ 0.20, remove the term.

The process would then stop or continue. It would stop if x4 x5 were not dropped, and otherwise, stepwise would continue to consider the significance of the next-to-last term, d1 d2 d3.

Specifying pe() rather than pr() switches to forward estimation. If you type

```
. stepwise, pe(.20): regress y1 x1 x2 (d1 d2 d3) (x4 x5)
```

stepwise performs forward-selection search. The logic for the first step is

1. Fit a model of y on nothing (meaning a constant).
2. Consider adding x1.
3. Consider adding x2.
4. Consider adding d1 d2 d3.
5. Consider adding x4 x5.
6. Find the term above that is most significant. If its significance level is < 0.20, add that term.

As with backward estimation, if you specify hierarchical,

```
. stepwise, pe(.20) hierarchical: regress y1 x1 x2 (d1 d2 d3) (x4 x5)
```

the search for the most significant term is restricted to the next term:

1. Fit a model of y on nothing (meaning a constant).
2. Consider adding x1—the first term.
3. If the significance is < 0.20, add the term.

If x1 were added, stepwise would next consider x2; otherwise, the search process would stop.

stepwise can also use a stepwise selection logic that alternates between adding and removing terms. The full logic for all the possibilities is given below.

Full search logic

Option	Logic
`pr()` (backward selection)	Fit the full model on all explanatory variables. While the least-significant term is “insignificant”, remove it and reestimate.
`pr() hierarchical` (backward hierarchical selection)	Fit full model on all explanatory variables. While the last term is “insignificant”, remove it and reestimate.
`pr() pe()` (backward stepwise)	Fit full model on all explanatory variables. If the least-significant term is “insignificant”, remove it and reestimate; otherwise, stop. Do that again: if the least-significant term is “insignificant”, remove it and reestimate; otherwise, stop. Repeatedly, if the most-significant excluded term is “significant”, add it and reestimate; if the least-significant included term is “insignificant”, remove it and reestimate; until neither is possible.
`pe()` (forward selection)	Fit “empty” model. While the most-significant excluded term is “significant”, add it and reestimate.
`pe() hierarchical` (forward hierarchical selection)	Fit “empty” model. While the next term is “significant”, add it and reestimate.
`pr() pe() forward` (forward stepwise)	Fit “empty” model. If the most-significant excluded term is “significant”, add it and reestimate; otherwise, stop. Do that again: if the most-significant excluded term is “significant”, add it and reestimate; otherwise, stop. Repeatedly, if the least-significant included term is “insignificant”, remove it and reestimate; if the most-significant excluded term is “significant”, add it and reestimate; until neither is possible.

Examples

The following two statements are equivalent; both include solely single-variable terms:

```
. stepwise, pr(.2): regress price mpg weight displ
. stepwise, pr(.2): regress price (mpg) (weight) (displ)
```

The following two statements are equivalent; the last term in each is `r1`, ..., `r4`:

```
. stepwise, pr(.2) hierarchical: regress price mpg weight displ (r1-r4)
. stepwise, pr(.2) hierarchical: regress price (mpg) (weight) (displ) (r1-r4)
```

To group variables `weight` and `displ` into one term, type

```
. stepwise, pr(.2) hierarchical: regress price mpg (weight displ) (r1-r4)
```

`stepwise` can be used with commands other than `regress`; for instance,

```
. stepwise, pr(.2): logit outcome (sex weight) treated1 treated2
. stepwise, pr(.2): logistic outcome (sex weight) treated1 treated2
```

Either statement would fit the same model because `logistic` and `logit` both perform logistic regression; they differ only in how they report results; see [R] **logit** and [R] **logistic**.

We use the `lockterm1` option to force the first term to be included in the model. To keep `treated1` and `treated2` in the model no matter what, we type

```
. stepwise, pr(.2) lockterm1: logistic outcome (treated1 treated2) ...
```

After `stepwise` estimation, we can type `stepwise` without arguments to redisplay results,

```
. stepwise
```
(*output from logistic appears*)

or type the underlying estimation command:

```
. logistic
```
(*output from logistic appears*)

At estimation time, we can specify options unique to the command being stepped:

```
. stepwise, pr(.2): logit outcome (sex weight) treated1 treated2, or
```

`or` is `logit`'s option to report odds ratios rather than coefficients; see [R] **logit**.

Estimation sample considerations

Whether you use backward or forward estimation, `stepwise` forms an estimation sample by taking observations with nonmissing values of all the variables specified (except for $depvar_1$ and $depvar_2$ for `intreg`). The estimation sample is held constant throughout the stepping. Thus if you type

```
. stepwise, pr(.2) hierarchical: regress amount sk edul sval
```

and variable `sval` is missing in half the data, that half of the data will not be used in the reported model, even if `sval` is not included in the final model.

The function `e(sample)` identifies the sample that was used. `e(sample)` contains 1 for observations used and 0 otherwise. For instance, if you type

```
. stepwise, pr(.2) pe(.10): logistic outcome x1 x2 (x3 x4) (x5 x6 x7)
```

and the final model is outcome on x1, x5, x6, and x7, you could re-create the final regression by typing

```
. logistic outcome x1 x5 x6 x7 if e(sample)
```

You could obtain summary statistics within the estimation sample of the independent variables by typing

```
. summarize x1 x5 x6 x7 if e(sample)
```

If you fit another model, e(sample) will automatically be redefined. Typing

```
. stepwise, lock pr(.2): logistic outcome (x1 x2) (x3 x4) (x5 x6 x7)
```

would automatically drop e(sample) and re-create it.

Messages

note: ________ dropped because of collinearity

Each term is checked for collinearity, and variables within the term are dropped if collinearity is found. For instance, say that you type

```
. stepwise, pr(.2): regress y x1 x2 (r1-r4) (x3 x4)
```

and assume that variables r1 through r4 are mutually exclusive and exhaustive dummy variables—perhaps r1, ..., r4 indicate in which of four regions the subject resides. One of the r1, ..., r4 variables will be automatically dropped to identify the model.

This message should cause you no concern.

Error message: between-term collinearity, variable ________

After removing any within-term collinearity, if stepwise still finds collinearity between terms, it refuses to continue. For instance, assume that you type

```
. stepwise, pr(.2): regress y1 x1 x2 (d1-d8) (r1-r4)
```

Assume that r1, ..., r4 identify in which of four regions the subject resides, and that d1, ..., d8 identify the same sort of information, but more finely. r1, say, amounts to d1 and d2; r2 to d3, d4, and d5; r3 to d6 and d7; and r4 to d8. You can estimate the d* variables or the r* variables, but not both.

It is your responsibility to specify noncollinear terms.

note: ________ dropped because of estimability
note: ________ obs. dropped because of estimability

You probably received this message in fitting a logistic or probit model. Regardless of estimation strategy, stepwise checks that the full model can be fit. The indicated variable had a 0 or infinite standard error.

For logistic, logit, and probit, this message is typically caused by one-way causation. Assume that you type

```
. stepwise, pr(.2): logistic outcome (x1 x2 x3) d1
```

and assume that variable d1 is an indicator (dummy) variable. Further assume that whenever d1 = 1, outcome = 1 in the data. Then the coefficient on d1 is infinite. One (conservative) solution to this problem is to drop the d1 variable and the d1==1 observations. The underlying estimation commands probit, logit, and logistic report the details of the difficulty and solution; stepwise simply accumulates such problems and reports the above summary messages. Thus if you see this message, you could type

```
. logistic outcome x1 x2 x3 d1
```

to see the details. Although you should think carefully about such situations, Stata's solution of dropping the offending variables and observations is, in general, appropriate.

Programming for stepwise

stepwise requires that *command_name* follow standard Stata syntax and allow the if qualifier; see **[U] 11 Language syntax**. 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 model degrees of freedom in e(df_m).

Stored results

stepwise stores whatever is stored by the underlying estimation command.

Also, stepwise stores stepwise in e(stepwise).

Methods and formulas

Some statisticians do not recommend stepwise procedures; see Sribney (1998) for a summary.

References

Afifi, A. A., S. May, and V. A. Clark. 2012. *Practical Multivariate Analysis*. 5th ed. Boca Raton, FL: CRC Press.

Beale, E. M. L. 1970. Note on procedures for variable selection in multiple regression. *Technometrics* 12: 909–914.

Bendel, R. B., and A. A. Afifi. 1977. Comparison of stopping rules in forward "stepwise" regression. *Journal of the American Statistical Association* 72: 46–53.

Berk, K. N. 1978. Comparing subset regression procedures. *Technometrics* 20: 1–6.

Draper, N., and H. Smith. 1998. *Applied Regression Analysis*. 3rd ed. New York: Wiley.

Efroymson, M. A. 1960. Multiple regression analysis. In *Mathematical Methods for Digital Computers*, ed. A. Ralston and H. S. Wilf, 191–203. New York: Wiley.

Gorman, J. W., and R. J. Toman. 1966. Selection of variables for fitting equations to data. *Technometrics* 8: 27–51.

Hocking, R. R. 1976. The analysis and selection of variables in linear regression. *Biometrics* 32: 1–49.

Hosmer, D. W., Jr., S. A. Lemeshow, and R. X. Sturdivant. 2013. *Applied Logistic Regression*. 3rd ed. Hoboken, NJ: Wiley.

Kennedy, W. J., Jr., and T. A. Bancroft. 1971. Model-building for prediction in regression based on repeated significance tests. *Annals of Mathematical Statistics* 42: 1273–1284.

Lindsey, C., and S. J. Sheather. 2010. Variable selection in linear regression. *Stata Journal* 10: 650–669.

Mantel, N. 1970. Why stepdown procedures in variable selection. *Technometrics* 12: 621–625.

——. 1971. More on variable selection and an alternative approach (letter to the editor). *Technometrics* 13: 455–457.

Sribney, W. M. 1998. FAQ: What are some of the problems with stepwise regression? http://www.stata.com/support/faqs/stat/stepwise.html.

Wang, Z. 2000. sg134: Model selection using the Akaike information criterion. *Stata Technical Bulletin* 54: 47–49. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 335–337. College Station, TX: Stata Press.

Williams, R. 2007. Stata tip 46: Step we gaily, on we go. *Stata Journal* 7: 272–274.

Also see

[R] **nestreg** — Nested model statistics