Most Stata commands allow the \texttt{by} prefix, which repeats the command for each group of observations for which the values of the variables in \textit{varlist} are the same. \texttt{by} without the \texttt{sort} option requires that the data be sorted by \textit{varlist}; see \cite{D} \texttt{sort}.

Stata commands that work with the \texttt{by} prefix indicate this immediately following their syntax diagram by reporting, for example, “\texttt{by} is allowed; see \cite{D} \texttt{by}” or “\texttt{bootstrap, by}, etc., are allowed; see \cite{U} 11.1.10 Prefix commands”.

\texttt{by} and \texttt{bysort} are really the same command; \texttt{bysort} is just \texttt{by} with the \texttt{sort} option.

The \texttt{varlist1} (\texttt{varlist2}) syntax is of special use to programmers. It verifies that the data are sorted by \texttt{varlist1} \texttt{varlist2} and then performs a \texttt{by} as if only \texttt{varlist1} were specified. For instance,

\begin{verbatim}
by pid (time): generate growth = (bp - bp[_n-1])/bp
\end{verbatim}

performs the \texttt{generate} by values of \texttt{pid} but first verifies that the data are sorted by \texttt{pid} and \texttt{time} within \texttt{pid}.

### Quick start

Generate \texttt{newv} as an observation number within each level of \texttt{catvar}

\begin{verbatim}
by catvar: generate newv = _n
\end{verbatim}

As above, but sort data by \texttt{catvar} first

\begin{verbatim}
by catvar, sort: generate newv = _n
\end{verbatim}

Same as above

\begin{verbatim}
bysort catvar: generate newv = _n
\end{verbatim}

As above, but sort by \texttt{v} within values of \texttt{catvar}

\begin{verbatim}
bysort catvar (v): generate newv = _n
\end{verbatim}

Generate \texttt{newv} as an observation number for each observation in levels of \texttt{catvar} and \texttt{v}

\begin{verbatim}
bysort catvar v: generate newv = _n
\end{verbatim}

Note: Any command that accepts the \texttt{by} prefix may be substituted for \texttt{generate} above.
by — Repeat Stata command on subsets of the data

Syntax

\[
\text{by} \varlist : \text{ stata\_cmd }
\]

\[
\text{bysort} \varlist : \text{ stata\_cmd }
\]

The above diagrams show by and bysort as they are typically used. The full syntax of the commands is

\[
\text{by} \varlist_1 [(\varlist_2)] [ , \text{ sort rc0} ] : \text{ stata\_cmd }
\]

\[
\text{bysort} \varlist_1 [(\varlist_2)] [ , \text{ rc0} ] : \text{ stata\_cmd }
\]

Options

sort specifies that if the data are not already sorted by \varlist, by should sort them.

rc0 specifies that even if the \text{ stata\_cmd} produces an error in one of the by-groups, then by is still to run the \text{ stata\_cmd} on the remaining by-groups. The default action is to stop when an error occurs. rc0 is especially useful when \text{ stata\_cmd} is an estimation command and some by-groups have insufficient observations.

Remarks and examples

Example 1

. use https://www.stata-press.com/data/r16/autornd
   (1978 Automobile Data)
   . keep in 1/20
   (54 observations deleted)
   . by mpg: egen mean_w = mean(weight)
   \text{not sorted}
   r(5);
   . sort mpg
   . by mpg: egen mean_w = mean(weight)
by — Repeat Stata command on subsets of the data

```
. list

<table>
<thead>
<tr>
<th>make</th>
<th>weight</th>
<th>mpg</th>
<th>mean_w</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buick Electra</td>
<td>4000</td>
<td>15</td>
<td>3916.667</td>
</tr>
<tr>
<td>Cad. Deville</td>
<td>4500</td>
<td>15</td>
<td>3916.667</td>
</tr>
<tr>
<td>Chev. Impala</td>
<td>3500</td>
<td>15</td>
<td>3916.667</td>
</tr>
<tr>
<td>AMC Pacer</td>
<td>3500</td>
<td>15</td>
<td>3916.667</td>
</tr>
<tr>
<td>Buick Riviera</td>
<td>4000</td>
<td>15</td>
<td>3916.667</td>
</tr>
<tr>
<td>Cad. Eldorado</td>
<td>4000</td>
<td>15</td>
<td>3916.667</td>
</tr>
<tr>
<td>AMC Concord</td>
<td>3000</td>
<td>20</td>
<td>3350</td>
</tr>
<tr>
<td>Chev. Malibu</td>
<td>3000</td>
<td>20</td>
<td>3350</td>
</tr>
<tr>
<td>Buick Skylark</td>
<td>3500</td>
<td>20</td>
<td>3350</td>
</tr>
<tr>
<td>Buick LeSabre</td>
<td>3500</td>
<td>20</td>
<td>3350</td>
</tr>
<tr>
<td>Buick Regal</td>
<td>3500</td>
<td>20</td>
<td>3350</td>
</tr>
<tr>
<td>Chev. Monte Carlo</td>
<td>3000</td>
<td>20</td>
<td>3350</td>
</tr>
<tr>
<td>Chev. Nova</td>
<td>3500</td>
<td>20</td>
<td>3350</td>
</tr>
<tr>
<td>Cad. Seville</td>
<td>4500</td>
<td>20</td>
<td>3350</td>
</tr>
<tr>
<td>AMC Spirit</td>
<td>2500</td>
<td>20</td>
<td>3350</td>
</tr>
<tr>
<td>Buick Century</td>
<td>3500</td>
<td>20</td>
<td>3350</td>
</tr>
<tr>
<td>Chev. Monza</td>
<td>3000</td>
<td>25</td>
<td>2500</td>
</tr>
<tr>
<td>Buick Opel</td>
<td>2000</td>
<td>25</td>
<td>2500</td>
</tr>
<tr>
<td>Chev. Chevette</td>
<td>2000</td>
<td>30</td>
<td>2000</td>
</tr>
<tr>
<td>Dodge Colt</td>
<td>2000</td>
<td>30</td>
<td>2000</td>
</tr>
</tbody>
</table>
```

by requires that the data be sorted. In the above example, we could have typed by mpg, sort: egen mean_w = mean(weight) or bysort mpg: egen mean_w = mean(weight) rather than the separate sort; all would yield the same results.

For more examples, see [U] 11.1.2 by varlist:, [U] 11.5 by varlist: construct, and [U] 13.7 Explicit subscripting. For extended introductions with detailed examples, see Cox (2002) and Mitchell (2010, chap. 7).

Technical note

by repeats the stata_cmd for each group defined by varlist. If stata_cmd stores results, only the results from the last group on which stata_cmd executes will be stored.

References


Also see

[D] sort — Sort data

[D] statsby — Collect statistics for a command across a by list

[P] byable — Make programs byable

[P] foreach — Loop over items

[P] forvalues — Loop over consecutive values

[P] while — Looping

[U] 11.1.2 by varlist:

[U] 11.1.10 Prefix commands

[U] 11.4 varname and varlists

[U] 11.5 by varlist: construct