**st_dropvar() — Drop variables or observations**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Remarks and examples</th>
<th>Conformability</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td>st_dropvar(transmorphic rowvector vars)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>void</td>
<td>st_dropobsin(real matrix range)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>void</td>
<td>st_dropobsif(real colvector select)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>void</td>
<td>st_keepvar(transmorphic rowvector vars)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>void</td>
<td>st_keepobsin(real matrix range)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>void</td>
<td>st_keepobsif(real colvector select)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**

void st_dropvar(transmorphic rowvector vars)
void st_dropobsin(real matrix range)
void st_dropobsif(real colvector select)
void st_keepvar(transmorphic rowvector vars)
void st_keepobsin(real matrix range)
void st_keepobsif(real colvector select)

**Description**

st_dropvar(vars) drops the variables specified. vars is a row vector that may contain either variable names or variable indices. st_dropvar(.) drops all variables and observations.

st_dropobsin() and st_dropobsif() have to do with dropping observations.

st_dropobsin(range) specifies the observations to be dropped:

- `st_dropobsin(5)` drops observation 5.
- `st_dropobsin((5,9))` drops observations 5 through 9.
- `st_dropobsin((5\8\12))` drops observations 5 and 8 and 12.
- `st_dropobsin((5,7\8,11\13,13))` drops observations 5 through 7, 8 through 11, and 13.
- `st_dropobsin(.)` drops all observations (but not the variables).
- `st_dropobsin(J(0,1,.))` drops no observations (or variables).

st_dropobsif(select) specifies a st_nobs() \times 1 vector. Observations i for which select_i \neq 0 are dropped.

st_keepvar(), st_keepobsin(), and st_keepobsif() do the same thing, except that the variables and observations to be kept are specified.
Remarks and examples

To drop all variables and observations, code any of the following:

\begin{verbatim}
st_dropvar(.)
st_keepvar(J(1,0,.))
st_keepvar(J(1,0,""))
\end{verbatim}

All do the same thing. Dropping all the variables clears the dataset.
Dropping all the observations, however, leaves the variables in place.

Conformability

\begin{verbatim}
st_dropvar(vars), st_keepvar(vars):
  vars:    1 × k
  result:  void

st_dropobsin(range), st_keepobsin(range):
  range:   k × 1 or k × 2
  result:  void

st_dropobsif(select), st_keepobsif(select):
  select:  st_nobs() × 1
  result:  void
\end{verbatim}

Diagnostics

\begin{verbatim}
\textbf{st_dropvar} (vars) and \textbf{st_keepvar} (vars) abort with error if any element of \texttt{vars} is missing unless \texttt{vars} is 1 × 1, in which case they drop or keep all the variables.

\textbf{st_dropvar} (vars) and \textbf{st_keepvar} (vars) abort with error if any element of \texttt{vars} is not a valid variable index or name, or if \texttt{vars} is a view. If \texttt{vars} is specified as names, abbreviations are not allowed.

\textbf{st_dropvar} () and \textbf{st_keepvar} () set \textbf{st_updata} () (see \textbf{[M-5] st_updata()}) unless all variables dropped are temporary; see \textbf{[M-5] st_tempname()}.\n
\textbf{st_dropobsin} (range) and \textbf{st_keepobsin} (range) abort with error if any element of \texttt{range} is missing unless \texttt{range} is 1 × 1, in which case they drop or keep all the observations.

\textbf{st_dropobsin} (range) and \textbf{st_keepobsin} (range) abort with error if any element of \texttt{range} is not a valid observation number (is not between 1 and \textbf{st_nobs} () [see \textbf{[M-5] st_nvar()]} inclusive) or if \texttt{range} is a view.

\textbf{st_dropobsif} (select) and \textbf{st_keepobsif} (select) abort with error if \texttt{select} is a view.

\textbf{st_dropobsin} (), \textbf{st_dropobsif} (), \textbf{st_keepobsin} (), and \textbf{st_keepobsif} () set \textbf{st_updata} () if any observations are removed from the data.
\end{verbatim}

Be aware that, after dropping any variables or observations, any previously constructed views (see \textbf{[M-5] st_view()}) are probably invalid because views are internally stored in terms of variable and observation numbers. Subsequent use of an invalid view may lead to unexpected results or an abort with error.
Also see

[M-4] stata — Stata interface functions