Description

recast changes the storage type of the variables identified in \texttt{varlist} to \texttt{type}.

Quick start

Recast numeric variable \texttt{v1} to type \texttt{double} from any other numeric type
\begin{verbatim}
recast double v1
\end{verbatim}

Recast string variable \texttt{v2} to \texttt{str30} from any length less than 30
\begin{verbatim}
recast str30 v2
\end{verbatim}

As above, but for length longer than 30
\begin{verbatim}
recast str30 v2, force
\end{verbatim}

Syntax

\begin{verbatim}
recast type varlist [, force]
\end{verbatim}

where \texttt{type} is \texttt{byte}, \texttt{int}, \texttt{long}, \texttt{float}, \texttt{double}, \texttt{str1}, \texttt{str2}, \ldots, \texttt{str2045}, or \texttt{strL}.

Option

\texttt{force} makes \texttt{recast} unsafe by causing the variables to be given the new storage type even if that will cause a loss of precision, introduction of missing values, or, for string variables, the truncation of strings.

\texttt{force} should be used with caution. \texttt{force} is for those instances where you have a variable saved as a \texttt{double} but would now be satisfied to have the variable stored as a \texttt{float}, even though that would lead to a slight rounding of its values.

Remarks and examples

See \texttt{[U] 12 Data} for a description of storage types. Also see \texttt{[D] compress} and \texttt{[D] destring} for alternatives to \texttt{recast}.
Example 1

recast refuses to change a variable’s type if that change is inappropriate for the values actually stored, so it is always safe to try:

. use https://www.stata-press.com/data/r16/auto
(1978 Automobile Data)
. describe headroom

<table>
<thead>
<tr>
<th>variable name</th>
<th>type</th>
<th>format</th>
<th>label</th>
<th>variable label</th>
</tr>
</thead>
<tbody>
<tr>
<td>headroom</td>
<td>float</td>
<td>%6.1f</td>
<td></td>
<td>Headroom (in.)</td>
</tr>
</tbody>
</table>

. recast int headroom
headroom: 37 values would be changed; not changed

Our attempt to change headroom from a float to an int was ignored—if the change had been made, 37 values would have changed. Here is an example where the type can be changed:

. describe mpg

<table>
<thead>
<tr>
<th>variable name</th>
<th>type</th>
<th>format</th>
<th>label</th>
<th>variable label</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpg</td>
<td>int</td>
<td>%8.0g</td>
<td></td>
<td>Mileage (mpg)</td>
</tr>
</tbody>
</table>

. recast byte mpg
. describe mpg

<table>
<thead>
<tr>
<th>variable name</th>
<th>type</th>
<th>format</th>
<th>label</th>
<th>variable label</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpg</td>
<td>byte</td>
<td>%8.0g</td>
<td></td>
<td>Mileage (mpg)</td>
</tr>
</tbody>
</table>

recast works with string variables as well as numeric variables, and it provides all the same protections:

. describe make

<table>
<thead>
<tr>
<th>variable name</th>
<th>type</th>
<th>format</th>
<th>label</th>
<th>variable label</th>
</tr>
</thead>
<tbody>
<tr>
<td>make</td>
<td>str18</td>
<td>%-18s</td>
<td></td>
<td>Make and Model</td>
</tr>
</tbody>
</table>

. recast str16 make
make: 2 values would be changed; not changed

recast can be used both to promote and to demote variables:

. recast str20 make
. describe make

<table>
<thead>
<tr>
<th>variable name</th>
<th>type</th>
<th>format</th>
<th>label</th>
<th>variable label</th>
</tr>
</thead>
<tbody>
<tr>
<td>make</td>
<td>str20</td>
<td>%-20s</td>
<td></td>
<td>Make and Model</td>
</tr>
</tbody>
</table>

Also see

[D] compress — Compress data in memory
[D] destring — Convert string variables to numeric variables and vice versa
[U] 12.2.2 Numeric storage types
[U] 12.4 Strings