**st_data() — Load copy of current Stata dataset**

<table>
<thead>
<tr>
<th>Description</th>
<th>Syntax</th>
<th>Remarks and examples</th>
<th>Conformability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Description**

`st_data(i, j)` returns the numeric value of the `i`th observation of the `j`th Stata variable. Observations are numbered 1 through `st_nobs()`. Variables are numbered 1 through `st_nvar()`.

`st_data(i, j)` is similar to `st_data(i, j)` except

1. `i` may be specified as a vector or matrix to obtain multiple observations simultaneously,
2. `j` may be specified using names or indices (indices are faster), and
3. `j` may be specified to obtain multiple variables simultaneously.

The net effect is that `st_data()` can return a scalar (the value of one variable in one observation), a row vector (the value of many variables in an observation), a column vector (the value of a variable in many observations), or a matrix (the value of many variables in many observations).

`st_data(i, j, selectvar)` works like `st_data(i, j)` except that only observations for which `selectvar ≠ 0` are returned.

`st_sdata()` and `st_sdata()` are the string variants of `st_data()` and `st_data()`. `st_sdata()` and `st_sdata()` are for use with numeric variables; they return missing (.) when used with string variables. `st_sdata()` and `st_sdata()` are for use with string variables; they return empty string ("") when used with numeric variables.

**Syntax**

```
real scalar  _st_data(real scalar i, real scalar j)
real matrix  st_data(real matrix i, rowvector j)
real matrix  st_data(real matrix i, rowvector j, scalar selectvar)
string scalar _st_sdata(real scalar i, real scalar j)
string matrix st_sdata(real matrix i, rowvector j)
string matrix st_sdata(real matrix i, rowvector j, scalar selectvar)
```

where

1. `i` may be specified as a `1 × 1` scalar, as a `1 × 1` scalar containing missing, as a column vector of observation numbers, as a row vector specifying an observation range, or as a `k × 2` matrix specifying both.
   a. `st_data(1, 2)` returns the first observation on the second variable.
b. `st_data(., 2)` returns all observations on the second variable.

c. `st_data((1\2\5), 2)` returns observations 1, 2, and 5 on the second variable.

d. `st_data((1,5), 2)` returns observations 1 through 5 on the second variable.

e. `st_data((1,5\7,9), 2)` returns observations 1 through 5 and observations 7 through 9 on the second variable.

When a range is specified, any element of the range \((i_1, i_2)\) may be specified to contribute zero observations if \(i_2 = i_1 - 1\).

2. \(j\) may be specified as a real row vector or as a string scalar or string row vector.

a. `st_data(., .)` returns the values of all variables, all observations of the Stata dataset.

b. `st_data(., 1)` returns the value of the first variable, all observations.

c. `st_data(., (3,1,9))` returns the values of the third, first, and ninth variables of all observations.

d. `st_data(., ("mpg", "weight"))` returns the values of variables mpg and weight, all observations.

e. `st_data(., ("mpg weight"))` does the same as d above.

f. `st_data(., ("gnp", "l.gnp"))` returns the values of gnp and the lag of gnp, all observations.

g. `st_data(., ("gnp l.gnp"))` does the same as f above.

h. `st_data(., ("mpg i.rep78"))` returns the value of mpg and the 5 pseudovariables associated with i.rep78. There are 5 pseudovariables because we are imagining that auto.dta is in memory; the actual number is a function of the values taken on by the variable in the sample specified. Factor variables can be specified only with string scalars; specifying ("mpg", "i.rep78") will not work.

3. `selectvar` may be specified as real or as a string. Observations for which `selectvar \neq 0` will be selected. If `selectvar` is real, it is interpreted as a variable number. If string, `selectvar` should contain the name of a Stata variable.

Specifying `selectvar` as "" or as missing (.) has the same result as not specifying `selectvar`; no observations are excluded.

Specifying `selectvar` as 0 means that observations with missing values of the variables specified by \(j\) are to be excluded.

**Remarks and examples**

Remarks are presented under the following headings:

- *Description of `st_data()` and `st_sdata()`*
- *Details of observation subscripting using `st_data()` and `st_sdata()`*
Description of _st_data() and _st_sdata()

_st_data() returns one variable’s value in one observation. You refer to variables and observations by their numbers. The first variable in the Stata dataset is 1; the first observation is 1.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_st_data(1, 1)</td>
<td>value of 1st obs., 1st variable</td>
</tr>
<tr>
<td>_st_data(1, 2)</td>
<td>value of 1st obs., 2nd variable</td>
</tr>
<tr>
<td>_st_data(2, 1)</td>
<td>value of 2nd obs., 1st variable</td>
</tr>
</tbody>
</table>

_st_sdata() works the same way. _st_data() is for use with numeric variables, and _st_sdata() is for use with string variables.

_st_data() and _st_sdata() are the fastest way to obtain the value of a variable in one observation.

Description of st_data() and st_sdata()

st_data() can be used just like _st_data(), and used that way, it produces the same result.

Variables, however, can be referred to by their names or their numbers:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>st_data(1, 1)</td>
<td>value of 1st obs., 1st variable</td>
</tr>
<tr>
<td>st_data(1, 2)</td>
<td>value of 1st obs., 2nd variable</td>
</tr>
<tr>
<td>st_data(2, 1)</td>
<td>value of 2nd obs., 1st variable</td>
</tr>
<tr>
<td>st_data(1, &quot;mpg&quot;)</td>
<td>value of 1st obs, variable mpg</td>
</tr>
<tr>
<td>st_data(2, &quot;mpg&quot;)</td>
<td>value of 2nd obs, variable mpg</td>
</tr>
</tbody>
</table>

Also, you may specify more than one variable:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>st_data(2, (1,2,3))</td>
<td>value of 2nd obs, variables 1, 2, and 3</td>
</tr>
<tr>
<td>st_data(2, (&quot;mpg&quot;,&quot;weight&quot;,&quot;displ&quot;))</td>
<td>value of 2nd obs, variables mpg, weight, and displ</td>
</tr>
<tr>
<td>st_data(2, &quot;mpg weight displ&quot;)</td>
<td>(same as previous)</td>
</tr>
</tbody>
</table>

Used this way, st_data() returns a row vector.

Similarly, you may obtain multiple observations:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>st_data((1\2\3), 10)</td>
<td>values of obs. 1, 2, and 3, variable 10</td>
</tr>
<tr>
<td>st_data((1,5), 10)</td>
<td>values of obs. 1 through 5, variable 10</td>
</tr>
<tr>
<td>st_data((1,5)(7,9), 10)</td>
<td>values of obs. 1 through 5 and 7 through 9, variable 10</td>
</tr>
</tbody>
</table>

_st_sdata() works the same way as st_data().
Details of observation subscripting using \texttt{st\_data()} and \texttt{st\_sdata()}

1. \texttt{i} may be specified as a scalar: the specified, single observation is returned. \texttt{i} must be between 1 and \texttt{st\_nobs()}; see [M-5] \texttt{st\_nvar()}. 

2. \texttt{i} may be specified as a scalar containing missing value: all observations are returned.

3. \texttt{i} may be specified as a column vector: the specified observations are returned. Each element of \texttt{i} must be between 1 and \texttt{st\_nobs()} or may be missing. Missing is interpreted as \texttt{st\_nobs()}. 

4. \texttt{i} may be specified as a \(1 \times 2\) row vector: the specified range of observations is returned; \((c_1, c_2)\) returns the \(c_2 - c_1 + 1\) observations \(c_1\) through \(c_2\). \(c_2 - c_1 + 1\) must evaluate to a number greater than or equal to 0. In general, \(c_1\) and \(c_2\) must be between 1 and \texttt{st\_nobs()}, but if \((c_2 - c_1 + 1) = 0\), then \(c_1\) may be between 1 and \texttt{st\_nobs()} + 1 and \(c_2\) may be between 0 and \texttt{st\_nobs()}. Regardless, \(c_1 == \cdot\) or \(c_2 == \cdot\) is interpreted as \texttt{st\_nobs()}. 

5. \texttt{i} may be specified as a \(k \times 2\) matrix: \(((1,5),(7,7),(20,30))\) specifies observations 1 through 5, 7, and 20 through 30.

Conformability

\begin{align*}
\texttt{st\_data(}i, j\texttt{)}, \texttt{st\_sdata(}i, j\texttt{)}: \\
i: & \quad 1 \times 1 \\
j: & \quad 1 \times 1 \\
\text{result:} & \quad 1 \times 1 \\
\texttt{st\_data(}i, j\texttt{)}, \texttt{st\_sdata(}i, j\texttt{)}: \\
i: & \quad n \times 1 \text{ or } n_2 \times 2 \\
j: & \quad 1 \times k \text{ or } 1 \times 1 \text{ containing } k \text{ elements when expanded} \\
\text{result:} & \quad n \times k \\
\texttt{st\_data(}i, j, \text{selectvar}\texttt{)}, \texttt{st\_sdata(}i, j, \text{selectvar}\texttt{)}: \\
i: & \quad n \times 1 \text{ or } n_2 \times 2 \\
j: & \quad 1 \times k \text{ or } 1 \times 1 \text{ containing } k \text{ elements when expanded} \\
\text{selectvar:} & \quad 1 \times 1 \\
\text{result:} & \quad (n - e) \times k, \text{ where } e \text{ is number of observations excluded by } \text{selectvar}
\end{align*}

Diagnostics

\begin{align*}
\texttt{st\_data(}i, j\texttt{)} & \text{ returns missing (.)} \text{ if } i \text{ or } j \text{ is out of range; it does not abort with error.} \\
\texttt{st\_sdata(}i, j\texttt{)} & \text{ returns "" if } i \text{ or } j \text{ is out of range; it does not abort with error.} \texttt{st\_data(}i, j\texttt{)} \text{ and } \texttt{st\_sdata(}i, j\texttt{)} \text{ abort with error if any element of } i \text{ or } j \text{ is out of range. } j \text{ may be specified as variable names or variable indices. If names are specified, abbreviations are allowed. If you do not want this and no factor variables nor time-series–operated variables are specified, use } \texttt{st\_varindex()} \text{ (see [M-5] \texttt{st\_varindex()}) to translate variable names into variable indices.}
\end{align*}
Also see

[M-5] **st_view()** — Make matrix that is a view onto current Stata dataset

[M-5] **st_store()** — Modify values stored in current Stata dataset

[M-4] **Stata** — Stata interface functions

[D] **putmata** — Put Stata variables into Mata and vice versa