

ssd — Making summary statistics data (sem only)
[Description](#)[Syntax](#)[Options](#)[Remarks and examples](#)[Stored results](#)[Also see](#)

Description

`ssd` allows you to enter SSD to fit SEMs and allows you to create SSD from original, raw data to publish or to send to others (and thus preserve participant confidentiality). Data created by `ssd` may be used with `sem` but not `gsem`.

Syntax

To enter summary statistics data (SSD), the commands are

```
ssd init varlist
```

```
ssd set [#] observations #
```

```
ssd set [#] means vector
```

```
ssd set [#] {variances|sd} vector
```

```
ssd set [#] {covariances|correlations} matrix
```

```
ssd addgroup varname (to add second group)
```

```
ssd addgroup (to add subsequent groups)
```

```
ssd unaddgroup # (to remove last group)
```

```
ssd status [#] (to review status)
```

To build SSD from raw data, the command is

```
ssd build varlist [, group(varname) clear]
```

To review the contents of SSD, the commands are

```
ssd status [#]
```

```
ssd describe
```

```
ssd list [#]
```

In an emergency (`ssd` will tell you when), you may use

```
ssd repair
```

In the above,

A *vector* can be any of the following:

1. A space-separated list of numbers, for example,

```
. ssd set means 1 2 3
```

2. `(stata) matname`

where *matname* is the name of a Stata $1 \times k$ or $k \times 1$ matrix, for example,

```
. ssd set variances (stata) mymeans
```

3. `(mata) matname`

where *matname* is the name of a Mata $1 \times k$ or $k \times 1$ matrix, for example,

```
. ssd set sd (mata) mymeans
```

A *matrix* can be any of the following:

1. A space-separated list of numbers corresponding to the rows of the matrix, with backslashes (`\`) between rows. The numbers are either the lower triangle and diagonal or the diagonal and upper triangle of the matrix, for example,

```
. ssd set correlations 1 \ .2 1 \ .3 .5 1
```

or

```
. ssd set correlations 1 .2 .3 \ 1 .5 \ 1
```

2. `(ltd) ## ...`

which is to say, a space-separated list of numbers corresponding to the lower triangle and diagonal of the matrix, without backslashes between rows, for example,

```
. ssd set correlations (ltd) 1 .2 1 .3 .5 1
```

3. `(dut) ## ...`

which is to say, a space-separated list of numbers corresponding to the diagonal and upper triangle of the matrix, without backslashes between rows, for example,

```
. ssd set correlations (dut) 1 .2 .3 1 .5 1
```

4. `(stata) matname`

where *matname* is the name of a Stata $k \times k$ symmetric matrix, for example,

```
. ssd set correlations (stata) mymat
```

5. `(mata) matname`

where *matname* is the name of a Mata $k \times k$ symmetric matrix, for example,

```
. ssd set correlations (mata) mymat
```

Options

`group(varname)` is for use with `ssd build`. It specifies that separate groups of summary statistics be produced for each value of *varname*.

`clear` is for use with `ssd build`. It specifies that it is okay to replace the data in memory with SSD even if the original dataset has not been saved since it was last changed.

Remarks and examples

[stata.com](http://www.stata.com)

See

[SEM] **intro 11** Fitting models with summary statistics data (sem only)

for an introduction, and see

[SEM] **example 2** Creating a dataset from published covariances
 [SEM] **example 19** Creating multiple-group summary statistics data
 [SEM] **example 25** Creating summary statistics data from raw data

A summary statistics dataset is different from a regular, raw Stata dataset. Be careful not to use standard Stata data-manipulation commands with SSD in memory. The commands include

```
generate
replace
merge
append
drop
set obs
```

to mention a few. You may, however, use `rename` to change the names of the variables.

The other data-manipulation commands can damage your summary statistics dataset. If you make a mistake and do use one of these commands, do not attempt to repair the data yourself. Let `ssd` repair your data by typing

```
. ssd repair
```

`ssd` is usually successful as long as variables or observations have not been dropped.

Every time you use `ssd`, even for something as trivial as describing or listing the data, `ssd` verifies that the data are not corrupted. If `ssd` finds that they are, it suggests you type `ssd repair`:

```
. ssd describe
SSD corrupt
The summary statistics data should [ssd
describes the problem]. The data may be fixable;
type ssd repair.
. ssd repair
(data repaired)
. ssd describe
(usual output appears)
```

In critical applications, we also recommend you digitally sign your summary statistics dataset:

```
. datasignature set
5:5(65336):3718404259:2275399871      (data signature set)
```

Then at any future time, you can verify the data are still just as they should be:

```
. datasignature confirm
(data unchanged since 30jun2016 15:32)
```

The data signature is a function of the variable names. If you rename a variable—something that is allowed—then the data signature will change:

```
. rename varname newname
. datasignature confirm
data have changed since 30jun2016 15:32
r(9);
```

In that case, you can re-sign the data:

```
. datasignature set, reset
5:5(71728):3718404259:2275399871      (data signature set)
```

Before re-signing, however, if you want to convince yourself that the data are unchanged except for the variable name, type `datasignature report`. It is the part of the signature in parentheses that has to do with the variable names. `datasignature report` will tell you what the new signature would be, and you can verify that the other components of the signature match.

See [D] [datasignature](#).

Stored results

`ssd describe` stores the following in `r()`:

Scalars

<code>r(N)</code>	number of observations (total across groups)
<code>r(k)</code>	number of variables (excluding group variable)
<code>r(G)</code>	number of groups
<code>r(complete)</code>	1 if complete, 0 otherwise
<code>r(complete_means)</code>	1 if complete means, 0 otherwise
<code>r(complete_covariances)</code>	1 if complete covariances, 0 otherwise

Macros

<code>r(v#)</code>	variable names (excluding group variable)
<code>r(groupvar)</code>	name of group variable (if there is one)

Also see

[SEM] [intro 11](#) — Fitting models with summary statistics data (sem only)

[D] [datasignature](#) — Determine whether data have changed

[SEM] [example 2](#) — Creating a dataset from published covariances

[SEM] [example 19](#) — Creating multiple-group summary statistics data

[SEM] [example 25](#) — Creating summary statistics data from raw data