

notes — Place notes in data

[Description](#)[Quick start](#)[Menu](#)[Syntax](#)[Remarks and examples](#)[Reference](#)[Also see](#)

Description

`notes` attaches notes to the dataset in memory. These notes become a part of the dataset and are saved when the dataset is saved and retrieved when the dataset is used; see [\[D\] save](#) and [\[D\] use](#). `notes` can be attached generically to the dataset or specifically to a variable within the dataset.

Quick start

Attach “My note about data” to current dataset

```
notes: My note about data
```

Add note “There is one note for v1” to v1

```
notes v1: There is one note for v1
```

Add note “A note was added to v2 on” and a time stamp for the note

```
notes v2: A note was added to v2 on TS
```

Add note “Data have changed” to the dataset

```
notes: Data have changed
```

Remove the first note from the dataset

```
notes drop _dta in 1
```

Renumber notes after removing a note from the dataset

```
notes renumber _dta
```

As above, but for a variable

```
notes renumber v1
```

List all notes

```
notes
```

List notes for the dataset but omit notes applied to variables

```
notes _dta
```

List only notes for variables

```
notes *
```

Search all notes for the word “check”

```
notes search check
```

Menu

notes (add)

Data > Variables Manager

notes list and notes search

Data > Data utilities > Notes utilities > List or search notes

notes replace

Data > Variables Manager

notes drop

Data > Variables Manager

notes renumber

Data > Data utilities > Notes utilities > Renumber notes

Syntax

Attach notes to dataset

```
notes [ varname ] : text
```

List all notes

```
notes
```

List specific notes

```
notes [ list ] varlist [ in # [ /# ] ]
```

Search for a text string across all notes in all variables and `_dta`

```
notes search [ sometext ]
```

Replace a note

```
notes replace varname in # : text
```

Drop notes

```
notes drop varlist [ in # [ /# ] ]
```

Renumber notes

```
notes renumber varname
```

where *varname* is `_dta` or a varname, *varlist* is a varlist that may contain the `_dta`, and # is a number or the letter l.

If *text* includes the letters TS surrounded by blanks, the TS is removed, and a time stamp is substituted in its place.

Remarks and examples

stata.com

Remarks are presented under the following headings:

- [How notes are numbered](#)
- [Attaching and listing notes](#)
- [Selectively listing notes](#)
- [Searching and replacing notes](#)
- [Deleting notes](#)
- [Warnings](#)
- [Video example](#)

How notes are numbered

Notes are numbered sequentially, with the first note being 1. Say the `myvar` variable has four notes numbered 1, 2, 3, and 4. If you type `notes drop myvar` in 3, the remaining notes will be numbered 1, 2, and 4. If you now add another note, it will be numbered 5. That is, notes are not renumbered and new notes are added immediately after the highest numbered note. Thus, if you now dropped notes 4 and 5, the next note added would be 3.

You can renumber notes by using `notes renumber`. Going back to when `myvar` had notes numbered 1, 2, and 4 after dropping note 3, if you typed `notes renumber myvar`, the notes would be renumbered 1, 2, and 3. If you added a new note after that, it would be numbered 4.

Attaching and listing notes

A note is nothing formal; it is merely a string of text reminding you to do something, cautioning you against something, or saying anything else you might feel like jotting down. People who work with real data invariably end up with paper notes plastered around their terminal saying things like, “Send the new sales data to Bob”, “Check the income variable in `salary95`; I don’t believe it”, or “The gender dummy was significant!” It would be better if these notes were attached to the dataset.

Adding a note to your dataset requires typing `note` or `notes` (they are synonyms), a colon (:), and whatever you want to remember. The note is added to the dataset currently in memory.

```
. note: Send copy to Bob once verified.
```

You can replay your notes by typing `notes` (or `note`) by itself.

```
. notes
_dta:
  1. Send copy to Bob once verified.
```

Once you resave your data, you can replay the note in the future, too. You add more notes just as you did the first:

```
. note: Mary wants a copy, too.
. notes
_dta:
  1. Send copy to Bob once verified.
  2. Mary wants a copy, too.
```

You can place time stamps on your notes by placing the word `TS` (in capitals) in the text of your note:

```
. note: TS merged updates from JJ&F
. notes
_dta:
  1. Send copy to Bob once verified.
  2. Mary wants a copy, too.
  3. 19 Apr 2019 15:38 merged updates from JJ&F
```

Notes may contain SMCL directives:

```
. use https://www.stata-press.com/data/r16/auto
(1978 Automobile Data)
. note: check reason for missing values in {cmd:rep78}
. notes
_dta:
  1. from Consumer Reports with permission
  2. check reason for missing values in rep78
```

The notes we have added so far are attached to the dataset generically, which is why Stata prefixes them with `_dta` when it lists them. You can attach notes to variables:

```
. note mpg: is the 44 a mistake? Ask Bob.
. note mpg: what about the two missing values?
. notes
_dta:
  1. Send copy to Bob once verified.
  2. Mary wants a copy, too.
  3. 19 Apr 2019 15:38 merged updates from JJ&F
mpg:
  1. is the 44 a mistake? Ask Bob.
  2. what about the two missing values?
```

Up to 9,999 generic notes can be attached to `_dta`, and another 9,999 notes can be attached to each variable.

Selectively listing notes

Typing `notes` by itself lists all the notes. In full syntax, `notes` is equivalent to typing `notes _all in 1/1`. Here are some variations:

<code>notes _dta</code>	list all generic notes
<code>notes mpg</code>	list all notes for variable <code>mpg</code>
<code>notes _dta mpg</code>	list all generic notes and <code>mpg</code> notes
<code>notes _dta in 3</code>	list generic note 3
<code>notes _dta in 3/5</code>	list generic notes 3–5
<code>notes mpg in 3/5</code>	list <code>mpg</code> notes 3–5
<code>notes _dta in 3/1</code>	list generic notes 3 through last

Searching and replacing notes

You had a bad day yesterday, and you want to recheck the notes that you added to your dataset. Fortunately, you always put a time stamp on your notes.

```
. notes search "29 Jan"
_dta:
  2. 29 Jan 2019 13:40 check reason for missing values in foreign
```

Good thing you checked. It is `rep78` that has missing values.

```
. notes replace _dta in 2: TS check reason for missing values in rep78
  (note 2 for _dta replaced)
. notes
_dta:
  1. from Consumer Reports with permission
  2. 30 Jan 2019 12:32 check reason for missing values in rep78
```

Deleting notes

`notes drop` works much like listing notes, except that typing `notes drop` by itself does not delete all notes; you must type `notes drop _all`. Here are some variations:

<code>notes drop _dta</code>	delete all generic notes
<code>notes drop _dta in 3</code>	delete generic note 3
<code>notes drop _dta in 3/5</code>	delete generic notes 3–5
<code>notes drop _dta in 3/1</code>	delete generic notes 3 through last
<code>notes drop mpg in 4</code>	delete mpg note 4

Warnings

- Notes are stored with the data, and as with other updates you make to the data, the additions and deletions are not permanent until you save the data; see [D] [save](#).
- The maximum length of one note is 67,784 characters for Stata/MP, Stata/SE, and Stata/IC.

Video example

[How to add notes to a variable](#)

Reference

Long, J. S. 2009. *The Workflow of Data Analysis Using Stata*. College Station, TX: Stata Press.

Also see

[D] [codebook](#) — Describe data contents

[D] [describe](#) — Describe data in memory or in file

[D] [ds](#) — Compactly list variables with specified properties

[D] [save](#) — Save Stata dataset

[D] [varmanage](#) — Manage variable labels, formats, and other properties

[U] [12.8 Characteristics](#)