Title

stata.com

describe — Describe data in memory or in a file

Description

`describe` produces a summary of the dataset in memory or of the data stored in a Stata-format dataset.

For a compact listing of variable names, use `describe, simple`.

Quick start

Describe all variables in the dataset

```
describe
```

Describe all variables starting with `code`

```
describe code*
```

Describe properties of the dataset

```
describe, short
```

Describe without abbreviating variable names

```
describe, fullnames
```

Create a dataset containing variable descriptions

```
describe, replace
```

Describe contents of `mydata.dta` without opening the dataset

```
describe using mydata
```

Menu

Data > Describe data > Describe data in memory or in a file

Syntax

Describe data in memory

```
describe [varlist] [, memory_options]
```

Describe data in a file

```
describe [varlist] using filename [, file_options]
```

memory_options	Description
simple	display only variable names
short	display only general information
fullnames	do not abbreviate variable names
numbers	display variable number along with name
replace	make dataset, not written report, of description
clear	for use with replace
varlist	store r(varlist) and r(sortlist) in addition to usual stored results; programmer's option

varlist does not appear in the dialog box.

file_options	Description
short	display only general information
simple	display only variable names
varlist	store r(varlist) and r(sortlist) in addition to usual stored results; programmer's option

varlist does not appear in the dialog box.

collect is allowed; see [U] 11.1.10 Prefix commands.

Options to describe data in memory

simple displays only the variable names in a compact format. simple may not be combined with other options.

short suppresses the specific information for each variable. Only the general information (number of observations, number of variables, and sort order) is displayed.

fullnames specifies that describe display the full names of the variables. The default is to present an abbreviation when the variable name is longer than 15 characters. describe using always shows the full names of the variables, so fullnames may not be specified with describe using.

numbers specifies that describe present the variable number with the variable name. If numbers is specified, variable names are abbreviated when the name is longer than eight characters. The numbers and fullnames options may not be specified together. numbers may not be specified with describe using.

`replace` and `clear` are alternatives to the options above. `describe` usually produces a written report, and the options above specify what the report is to contain. If you specify `replace`, however, no report is produced; the data in memory are instead replaced with data containing the information that the report would have presented. Each observation of the new data describes a variable in the original data; see *describe, replace* below.

`clear` may be specified only when `replace` is specified. `clear` specifies that the data in memory be cleared and replaced with the description information, even if the original data have not been saved to disk.

The following option is available with `describe` but is not shown in the dialog box:

`varlist`, an option for programmers, specifies that, in addition to the usual stored results, `r(varlist)` and `r(sortlist)` be stored, too. `r(varlist)` will contain the names of the variables in the dataset. `r(sortlist)` will contain the names of the variables by which the data are sorted.

Options to describe data in a file

`short` suppresses the specific information for each variable. Only the general information (number of observations, number of variables, and sort order) is displayed.

`simple` displays only the variable names in a compact format. `simple` may not be combined with other options.

The following option is available with `describe` but is not shown in the dialog box:

`varlist`, an option for programmers, specifies that, in addition to the usual stored results, `r(varlist)` and `r(sortlist)` be stored, too. `r(varlist)` will contain the names of the variables in the dataset. `r(sortlist)` will contain the names of the variables by which the data are sorted.

Because Stata/MP and Stata/SE can create truly large datasets, there might be too many variables in a dataset for their names to be stored in `r(varlist)`, given the current maximum length of macros, as determined by `set maxvar`. Should that occur, `describe using` will issue the error message "too many variables", r(103).

Remarks and examples

stata.com

Remarks are presented under the following headings:

describe

If `describe` is typed with no operands, the contents of the dataset currently in memory are described.

The *varlist* in the `describe using` syntax differs from standard Stata varlists in two ways. First, you cannot abbreviate variable names; that is, you have to type `displacement` rather than `displ`. However, you can use the abbreviation character (~) to indicate abbreviations, for example, `displ~`. Second, you may not refer to a range of variables; specifying `price-trunk` is considered an error.

If you are using `frames` to work with multiple datasets in memory, you can use `frames describe` to describe data from one or more frames. However, you might also want to create alias variables, which is similar to copying variables across frames but is more memory efficient. When the dataset in memory contains alias variables, `describe` tries to report the storage type of the linked variable. If an alias variable's linkage is broken, then `describe` will report `unknown` for the storage type. In either case, the storage type text will be a clickable link that runs command `fralias describe` on the associated variable. For examples of `describe` output and behavior with alias variables, see [D] **fralias**.

For alias variables in *filename*, `describe using` reports `alias` for the storage type.

▷ Example 1

The basic description includes some general information on the number of variables and observations, along with a description of every variable in the dataset:

```
. use https://www.stata-press.com/data/r18/states
(State data)

. describe, numbers

Contains data from https://www.stata-press.com/data/r18/states.dta
 Observations:            50                  State data
    Variables:             5                  3 Jan 2022 15:17
                                              (_dta has notes)
-------------------------------------------------------------------------------
    Variable  Storage   Display    Value
        name     type    format    label      Variable label
-------------------------------------------------------------------------------
    1. state     str8    %9s
    2. region    int     %8.0g      reg        Census Region
    3. median~e  float   %9.0g                 Median Age
    4. marria~e  long    %12.0g                Marriages per 100,000
    5. divorc~e  long    %12.0g                Divorces per 100,000
-------------------------------------------------------------------------------
Sorted by: region
```

In this example, the dataset in memory comes from the file `states.dta` and contains 50 observations on 5 variables. The dataset is labeled "State data" and was last modified on January 3, 2020, at 15:17 (3:17 p.m.). The "_dta has notes" message indicates that a note is attached to the dataset; see [U] **12.7 Notes attached to data**.

The first variable, `state`, is stored as a `str8` and has a display format of `%9s`.

The next variable, `region`, is stored as an `int` and has a display format of `%8.0g`. This variable has associated with it a *value label* called `reg`, and the variable is labeled `Census Region`.

The third variable, which is abbreviated `median~e`, is stored as a `float`, has a display format of `%9.0g`, has no value label, and has a variable label of `Median Age`. The variables that are abbreviated `marria~e` and `divorc~e` are both stored as `long`s and have display formats of `%12.0g`. These last two variables are labeled `Marriages per 100,000` and `Divorces per 100,000`, respectively.

The data are sorted by `region`.

Because we specified the `numbers` option, the variables are numbered; for example, `region` is variable 2 in this dataset.

◁

▷ Example 2

To view the full variable names, we could omit the `numbers` option and specify the `fullnames` option.

```
. describe, fullnames
Contains data from https://www.stata-press.com/data/r18/states.dta
 Observations:            50                  State data
    Variables:             5                  3 Jan 2022 15:17
                                              (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
state	str8	%9s		
region	int	%8.0g	reg	Census Region
median_age	float	%9.0g		Median Age
marriage_rate	long	%12.0g		Marriages per 100,000
divorce_rate	long	%12.0g		Divorces per 100,000

```
Sorted by: region
```

Here we did not need to specify the `fullnames` option to see the unabbreviated variable names because the longest variable name is 13 characters. Omitting the `numbers` option results in 15-character variable names being displayed.

◁

❑ Technical note

The output from `describe` allows you to compute the size of the dataset. If you are curious, you can compute it for this dataset as follows:

$$(8 + 2 + 4 + 4 + 4) \times 50 = 1100$$

The numbers 8, 2, 4, 4, and 4 are the storage requirements for a `str8`, `int`, `float`, `long`, and `long`, respectively; see **[U] 12.2.2 Numeric storage types**. Fifty is the number of observations in the dataset.

❑

▷ Example 3

If we specify the `short` option, only general information about the data is presented:

```
. describe, short
Contains data from https://www.stata-press.com/data/r18/states.dta
 Observations:            50                  State data
    Variables:             5                  3 Jan 2022 15:17
Sorted by: region
```

◁

If we specify a *varlist*, only the variables in that *varlist* are described.

Example 4

Let's change datasets. The `describe` *varlist* command is particularly useful when combined with the '`*`' wildcard character. For instance, we can describe all the variables whose names start with `pop` by typing `describe pop*`:

```
. use https://www.stata-press.com/data/r18/census
(1980 Census data by state)
. describe pop*
```

Variable name	Storage type	Display format	Value label	Variable label
pop	long	%12.0gc		Population
poplt5	long	%12.0gc		Pop, < 5 year
pop5_17	long	%12.0gc		Pop, 5 to 17 years
pop18p	long	%12.0gc		Pop, 18 and older
pop65p	long	%12.0gc		Pop, 65 and older
popurban	long	%12.0gc		Urban population

We can `describe` the variables `state`, `region`, and `pop18p` by specifying them:

```
. describe state region pop18p
```

Variable name	Storage type	Display format	Value label	Variable label
state	str14	%-14s		State
region	int	%-8.0g	cenreg	Census region
pop18p	long	%12.0gc		Pop, 18 and older

◁

Typing `describe using` *filename* describes the data stored in *filename*. If an extension is not specified, `.dta` is assumed.

Example 5

We can `describe` the contents of `states.dta` without disturbing the data that we currently have in memory by typing

```
. describe using https://www.stata-press.com/data/r18/states
Contains data                                  State data
 Observations:            50                   3 Jan 2022 15:17
    Variables:             5
```

Variable name	Storage type	Display format	Value label	Variable label
state	str8	%9s		
region	int	%8.0g	reg	Census Region
median_age	float	%9.0g		Median Age
marriage_rate	long	%12.0g		Marriages per 100,000
divorce_rate	long	%12.0g		Divorces per 100,000

```
Sorted by: region
```

◁

describe, replace

`describe` with the `replace` option is rarely used, although you may sometimes find it convenient.

Think of `describe, replace` as separate from but related to `describe` without the `replace` option. Rather than producing a written report, `describe, replace` produces a new dataset that contains the same information a written report would. For instance, try the following:

```
. sysuse auto, clear
. describe
(report appears; data in memory unchanged)
. list
(visual proof that data are unchanged)
. describe, replace
(no report appears, but the data in memory are changed!)
. list
(visual proof that data are changed)
```

`describe, replace` changes the original data in memory into a dataset containing an observation for each variable in the original data. Each observation in the new data describes a variable in the original data. The new variables are

1. `position`, a variable containing the numeric position of the original variable (1, 2, 3, ...).
2. `name`, a variable containing the name of the original variable, such as `"make"`, `"price"`, `"mpg"`, ...
3. `type`, a variable containing the storage type of the original variable, such as `"str18"`, `"int"`, `"float"`, ...
4. `isnumeric`, a variable equal to 1 if the original variable was numeric and equal to 0 if it was string.
5. `format`, a variable containing the display format of the original variable, such as `"%-18s"`, `"%8.0gc"`, ...
6. `vallab`, a variable containing the name of the value label associated with the original variable, if any.
7. `varlab`, a variable containing the variable label of the original variable, such as `"Make and model"`, `"Price"`, `"Mileage (mpg)"`, ...

In addition, the data contain the following characteristics:

`_dta[d_filename]`, the name of the file containing the original data.

`_dta[d_filedate]`, the date and time the file was written.

`_dta[d_N]`, the number of observations in the original data.

`_dta[d_sortedby]`, the variables on which the original data were sorted, if any.

Stored results

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

Scalars

`r(N)`	number of observations
`r(k)`	number of variables
`r(width)`	width of dataset
`r(changed)`	flag indicating data have changed since last saved

Macros

`r(datalabel)`	dataset label
`r(varlist)`	variables in dataset (if `varlist` specified)
`r(sortlist)`	variables by which data are sorted (if `varlist` specified)

`describe, replace` stores nothing in `r()`.

References

Cox, N. J. 2015. Speaking Stata: A set of utilities for managing missing values. *Stata Journal* 15: 1174–1185.

Dietz, T., and L. Kalof. 2009. *Introduction to Social Statistics: The Logic of Statistical Reasoning*. Chichester, UK: Wiley.

Also see

[D] **ds** — Compactly list variables with specified properties

[D] **varmanage** — Manage variable labels, formats, and other properties

[D] **cf** — Compare two datasets

[D] **codebook** — Describe data contents

[D] **compare** — Compare two variables

[D] **compress** — Compress data in memory

[D] **format** — Set variables' output format

[D] **fralias** — Alias variables from linked frames

[D] **label** — Manipulate labels

[D] **lookfor** — Search for string in variable names and labels

[D] **notes** — Place notes in data

[D] **order** — Reorder variables in dataset

[D] **rename** — Rename variable

[SVY] **svydescribe** — Describe survey data

[U] 6 Managing memory