Title

stata.com

table intro — Introduction to tables of frequencies, summaries, and command results

Description Remarks and examples Reference Also see

Description

Tables allow us to effectively communicate information about our data and results from our analyses. The `table` command is a flexible tool for creating tables. It allows you to create tabulations, tables of summary statistics, tables of results from hypothesis tests, tables of regression results, and more. `table` also allows you to customize the table so that it effectively communicates the results.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Overview

The `table` command allows you to create various types of tables.

1. Tabulations of one, two, or more categorical variables, reporting frequencies, percentages, or proportions.
2. Tables of summary statistics such as means, standard deviations, medians, and the like—perhaps computed across levels of one or more categorical variables.
3. Tables of results from other Stata commands, such as regression results or results of classic hypothesis tests.
4. Tables combining summary statistics and results from other Stata commands.

`table` obtains the statistics it reports in two ways. The `table` command itself can compute summary statistics. `table` also provides a `command()` option that allows you to run any Stata command and include its results in the table.

`table` also provides much flexibility for you to arrange the results in your table in a meaningful way. You can control how the rows, columns, and potentially even separate tables are defined. For instance, you might type

```
. table (var1) (var2) ...
```

to place the levels of `var1` on the rows and the levels of `var2` on the columns. The first set of parentheses is used to define the rows, and the second set is used to define the columns. The parentheses could be omitted in this case, but for clarity we will use them in our discussion here.

We could use the levels of both var1 and var2 to define the rows.

```
. table (var1 var2) ...
```

If we have an additional variable, var3, we can define rows by both var1 and var2 and the columns by var3 by typing

```
. table (var1 var2) (var3) ...
```

or we can place the levels of var2 and var3 on the columns,

```
. table (var1) (var2 var3) ...
```

We could even create separate tables for the levels of var3,

```
. table (var1) (var2) (var3) ...
```

The third set of parentheses defines the separate tables.

If our table reports multiple statistics, say, means and standard deviations, we can use the keyword result to specify how they are included in the table layout. To place the statistics on separate rows, we could type

```
. table (var1 result) (var2) ...
```

The flexibility of the table layout goes beyond these examples. You can add additional variables to define rows, columns, and tables as is appropriate for the table you wish to create.

In addition to controlling the layout of your table, you can customize the results by specifying formats, including stars representing significance, and selecting from styles that determine how the headers of the table are displayed.

Tabulations

Tabulations allow you to examine the distribution of your data across the levels of one or more categorical variables. Tabulations often report frequencies, percentages, and proportions.

With table, you can easily create one-way tabulations, two-way tabulations, three-way tabulations, and even more complex tabulations. For example, we could create a two-way tabulation of region and diabetes, reporting frequencies,

```
. table (region) (diabetes)
```

	Diabetes status		
	Not diabetic	Diabetic	Total
Region			
NE	1,997	98	2,095
MW	2,648	125	2,773
S	2,692	161	2,853
W	2,513	115	2,628
Total	9,850	499	10,349

or we can report percentages,

```
. table (region) (diabetes), statistic(percent)
```

	Diabetes status		
	Not diabetic	Diabetic	Total
Region			
NE	19.30	0.95	20.24
MW	25.59	1.21	26.79
S	26.01	1.56	27.57
W	24.28	1.11	25.39
Total	95.18	4.82	100.00

To learn how to create tabulations, see

[R] **table oneway**	One-way tabulation
[R] **table twoway**	Two-way tabulation
[R] **table multiway**	Multiway tables

In these entries, we provide simplified syntax for creating the specific type of tabulation you are interested in, and we provide a number of examples demonstrating how to build these tables and customize the results.

Tables of summary statistics

`table` can compute summary statistics such as minimums, maximums, means, standard deviations, medians, and interquartile ranges. Summary statistics can be computed for one or more variables. Summary statistics can also be computed separately for groups of data. For example, we can create a table reporting the means of `age`, `height`, and `bmi`,

```
. table, statistic(mean age height bmi)
```

Age (years)	47.57965
Height (cm)	167.6509
Body mass index (BMI)	25.5376

or we can compute these means separately for males and females,

```
. table () (sex), statistic(mean age height bmi)
```

	Sex		
	Male	Female	Total
Age (years)	47.4238	47.72057	47.57965
Height (cm)	174.7421	161.2393	167.6509
Body mass index (BMI)	25.50999	25.56256	25.5376

To learn how to create tables of summary statistics, including the simplified `table` syntax and worked examples, see

[R] **table summary**	Table of summary statistics

You can also create a table of summary statistics with `dtable`. The advantage of using `dtable` is that you can create a table with a title and notes and export it to a variety of file types, all in a single step. Unlike `table`, however, `dtable` has a predefined layout, and you will not be able to specify how the results should be arranged in the table.

Tables of results from other commands

With `table`'s `command()` option, you can run any Stata command and place statistics reported by that command in your table. For instance, you might use `test` to test for differences in means or use `regress` to fit a linear regression model. You can include any stored results from these commands in your table. For example, we can create a table to compare coefficients from linear regressions of `bpsysol` on `age` and `weight` fit separately for males and females.

```
. table (colname) (sex result), command(regress bpsystol age weight)
```

	Sex		
	Male	Female	Total
Age (years)	.4789361	.7735499	.6379892
Weight (kg)	.3346106	.4586108	.4069041
Intercept	84.08037	61.70456	71.27096

To learn how to create tables with results from other commands, including simplified `table` syntax and worked examples, see

[R] **table hypothesis tests**	Table of hypothesis tests
[R] **table regression**	Table of regression results

Flexible tables combining results

`table` can create tables with combinations of frequencies, other summary statistics, and results from other table commands. To see the full syntax and discussion of how `table` automates the table layout, see

[R] **table**	Table of frequencies, summaries, and command results

Formatting, customizing, and exporting tables

`table` allows you to customize your results by specifying the layout and with options that change the numeric formatting, add characters such as percent signs or parentheses to specific types of statistics, add stars representing significance levels, and modify the look of confidence intervals. In addition, Stata provides predefined styles that you may select from using the `style()` option. These styles control which labels are displayed in the headers of the tables, how the labels are aligned, how the statistics are aligned within the cells, and more. To learn about the predefined styles that you can select from, see [TABLES] **Predefined styles**. Examples of using the `style()` and other formatting options are provided in the entries for specific types of tables listed in the previous sections.

The results from `table` can be customized even beyond what its formatting options allow. `table` is unique in that it stores its results in a format that we refer to as a "collection". Stata's `collect` suite of commands can be used to produce highly customized tables from results in a collection and to export those tables to presentation-ready formats such as HTML, Word, LaTeX, PDF, Excel, and more. Examples of using `collect` to modify labels and table layout are also provided in the entries for the specific types of tables listed in the previous sections. To learn more about the `collect` commands, see [TABLES] **Intro** and the entries discussed therein.

Reference

Huber, C. 2021. Customizable tables in Stata 17, part 1: The new table command. *The Stata Blog: Not Elsewhere Classified*. https://blog.stata.com/2021/06/07/customizable-tables-in-stata-17-part-1-the-new-table-command/.

Also see

[R] **table** — Table of frequencies, summaries, and command results

[TABLES] **Intro** — Introduction