Example 3 — Table of comparative summary statistics

Description Remarks and examples Reference Also see

Description

In this example, we demonstrate how to use `table` to compute summary statistics for levels of a categorical variable and store them in a collection. We also demonstrate how to use the `collect` suite of commands to create a customized table with these results.

Remarks and examples

Remarks are presented under the following headings:

Computing statistics with the table command

Below, we use data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981). We want to create a table to compare summary statistics for males and females.

With the `table` command, we can compute several types of summary statistics. Below, we use the `statistic()` option to compute the `mean` and standard deviation (`sd`) of `age`, body mass index (`bmi`), and systolic blood pressure (`bpsystol`) for each category of `sex`. We place our variables (`var`) on the rows and the levels of `sex` on the columns. Additionally, we format the means and standard deviations to display only two digits to the right of the decimal.

```
. use https://www.stata-press.com/data/r19/nhanes2l
(Second National Health and Nutrition Examination Survey)
. table (var) (sex),
>     statistic(mean age bmi bpsystol)
>     statistic(sd age bmi bpsystol)
>     nformat(%6.2f)
```

	Sex		
	Male	Female	Total
Age (years)			
Mean	47.42	47.72	47.58
Standard deviation	17.17	17.26	17.21
Body mass index (BMI)			
Mean	25.51	25.56	25.54
Standard deviation	4.02	5.60	4.91
Systolic blood pressure			
Mean	132.89	129.07	130.88
Standard deviation	20.99	25.13	23.33

The table above reports summary statistics for continuous variables. We might also want to incorporate statistics for categorical variables. For instance, let's report frequencies and percentages for the levels of `diabetes` and `hlthstat`. The statistic `fvfrequency` provides the frequency for each level of a categorical variable, and `fvpercent` reports the percentage of observations in each category. We still want to format our means and standard deviations but not our other statistics. With `nformat()`, we can specify the statistics to which we want to apply the format.

```
. table (var) (sex),
>     statistic(fvfrequency diabetes) statistic(fvpercent diabetes)
>     statistic(mean age bmi) statistic(sd age bmi)
>     statistic(fvfrequency hlthstat) statistic(fvpercent hlthstat)
>     statistic(mean bpsystol) statistic(sd bpsystol)
>     nformat(%6.2f mean sd)
```

	Sex		
	Male	Female	Total
Diabetes status=Not diabetic			
Factor-variable frequency	4,698	5,152	9,850
Factor-variable percent	95.58	94.81	95.18
Diabetes status=Diabetic			
Factor-variable frequency	217	282	499
Factor-variable percent	4.42	5.19	4.82
Age (years)			
Mean	47.42	47.72	47.58
Standard deviation	17.17	17.26	17.21
Body mass index (BMI)			
Mean	25.51	25.56	25.54
Standard deviation	4.02	5.60	4.91
Health status=Excellent			
Factor-variable frequency	1,252	1,155	2,407
Factor-variable percent	25.50	21.29	23.29
Health status=Very good			
Factor-variable frequency	1,213	1,378	2,591
Factor-variable percent	24.71	25.40	25.07
Health status=Good			
Factor-variable frequency	1,340	1,598	2,938
Factor-variable percent	27.30	29.45	28.43
Health status=Fair			
Factor-variable frequency	722	948	1,670
Factor-variable percent	14.71	17.47	16.16
Health status=Poor			
Factor-variable frequency	382	347	729
Factor-variable percent	7.78	6.40	7.05
Systolic blood pressure			
Mean	132.89	129.07	130.88
Standard deviation	20.99	25.13	23.33

We now have a table with summary statistics for males and females in our data. However, we likely want to polish the table so that the labels are not distracting.

Customizing the table

By default, `table` will display the table and store the results in a collection called `Table`. We can now use the `collect` suite of commands to work with this collection and modify the look of the table.

To get started, note that the statistics are stored as levels of the dimension `result`. We can see the levels of this dimension by using `collect levelsof`. We will use the names of the dimension and its levels in the `collect` subcommands that we use to modify our table.

```
. collect levelsof result
Collection: Table
 Dimension: result
    Levels: fvfrequency fvpercent mean sd
```

First, let's remove the labels for the statistics in the row headers. We can use `collect style header` to hide the level labels for the dimension `result`. Then, we preview our table with `collect preview`.

```
. collect style header result, level(hide)
. collect preview
```

	Sex		
	Male	Female	Total
Diabetes status=Not diabetic	4,698	5,152	9,850
	95.58	94.81	95.18
Diabetes status=Diabetic	217	282	499
	4.42	5.19	4.82
Age (years)	47.42	47.72	47.58
	17.17	17.26	17.21
Body mass index (BMI)	25.51	25.56	25.54
	4.02	5.60	4.91
Health status=Excellent	1,252	1,155	2,407
	25.50	21.29	23.29
Health status=Very good	1,213	1,378	2,591
	24.71	25.40	25.07
Health status=Good	1,340	1,598	2,938
	27.30	29.45	28.43
Health status=Fair	722	948	1,670
	14.71	17.47	16.16
Health status=Poor	382	347	729
	7.78	6.40	7.05
Systolic blood pressure	132.89	129.07	130.88
	20.99	25.13	23.33

The variable labels and value labels for our categorical variables are bound by an equal sign. Instead of repeating the variable labels, we can use `collect style row stack` to list each one only once and stack these headers in a single column. We also specify the `spacer` option to insert a blank line between row dimensions. Finally, we can remove the border on the right side of the row headers by setting the border pattern to `nil`. We then preview our table once more.

```
. collect style row stack, nobinder spacer
. collect style cell border_block, border(right, pattern(nil))
. collect preview
```

	Sex		
	Male	Female	Total
Diabetes status			
Not diabetic	4,698	5,152	9,850
	95.58	94.81	95.18
Diabetic	217	282	499
	4.42	5.19	4.82
Age (years)	47.42	47.72	47.58
	17.17	17.26	17.21
Body mass index (BMI)	25.51	25.56	25.54
	4.02	5.60	4.91
Health status			
Excellent	1,252	1,155	2,407
	25.50	21.29	23.29
Very good	1,213	1,378	2,591
	24.71	25.40	25.07
Good	1,340	1,598	2,938
	27.30	29.45	28.43
Fair	722	948	1,670
	14.71	17.47	16.16
Poor	382	347	729
	7.78	6.40	7.05
Systolic blood pressure	132.89	129.07	130.88
	20.99	25.13	23.33

This layout is one nice way to compare the summary statistics. We could continue to modify its style to finalize our table.

However, we may also want to consider another layout—one in which the means of continuous variables and frequencies of categorical variables are in one column and the standard deviations of continuous variables and percentages for categorical variables are in another column.

Currently, the frequencies for the categorical variables are tagged with the level `fvfrequency` of the `result` dimension, and the percentages are tagged with level `fvpercent` of the `result` dimension. To align the frequencies with the means and the percentages with the standard deviations, we recode them to the levels `mean` and `sd` of the same dimension. Then, we lay out our table with the variables on the rows and the results for males and females on the columns. Note that by typing `sex[1 2]`, we specify that only levels `1` and `2` of the `sex` dimension be included. This allows us to omit the statistics that `table` computed for all observations in the data and that would be included if we simply include the dimension `sex`.

```
. collect recode result fvfrequency=mean  fvpercent=sd
(42 items recoded in collection Table)
. collect layout (var) (sex[1 2]#result)
Collection: Table
      Rows: var
   Columns: sex[1 2]#result
   Table 1: 15 x 4
```

	Sex			
	Male		Female	
Diabetes status				
Not diabetic	4698.00	95.58	5152.00	94.81
Diabetic	217.00	4.42	282.00	5.19
Age (years)	47.42	17.17	47.72	17.26
Body mass index (BMI)	25.51	4.02	25.56	5.60
Health status				
Excellent	1252.00	25.50	1155.00	21.29
Very good	1213.00	24.71	1378.00	25.40
Good	1340.00	27.30	1598.00	29.45
Fair	722.00	14.71	948.00	17.47
Poor	382.00	7.78	347.00	6.40
Systolic blood pressure	132.89	20.99	129.07	25.13

Now, we can finish customizing this table by adding percent signs to the percentages, enclosing our standard deviations in parentheses, and fixing the numeric formatting. (Now that the frequencies are part of the level `mean`, they have the numeric format that we applied earlier to that level.)

We can use `collect style cell` to modify all cells in the table or specific cells.

First, we add a percent sign to our percentages. Because we recoded the percentages to the `sd` level of `result`, we will need to refer to them with the tag `result[sd]`. However, this is not enough. If we refer to only `result[sd]`, we will refer to both standard deviations and percentages. To apply a change only to our categorical variables, we type `result[sd]#var[i.diabetes i.hlthstat]`. By interacting these two tags, we reference only values that are tagged with the `sd` level of `result` as well as the levels of either `i.diabetes` or `i.hlthstat` of `var`.

The option `sformat()` changes the string format, and `%s` refers to the numeric value. The text will be placed around our numeric values in the table as we place it around `%s` in this option. Adding a percent sign requires a special character, `%%`.

Similarly, we can type `result[sd]#var[age bmi bpsystol]` to refer to the standard deviations of our continuous variables. We enclose these values in parentheses.

```
. collect style cell result[sd]#var[i.diabetes i.hlthstat], sformat("%s%%")
. collect style cell result[sd]#var[age bmi bpsystol], sformat("(%s)")
```

Last, we do not want to display any digits to the right of the decimal for the frequencies. So we use `collect style cell` with the `nformat()` option for the frequencies (tagged with `mean` of the `result` dimension and of the levels `hlthstat` and `diabetes` of the `var` dimension).

```
. collect style cell result[mean]#var[i.diabetes i.hlthstat], nformat(%4.0f)
. collect preview
```

	Sex			
	Male		Female	
Diabetes status				
Not diabetic	4698	95.58%	5152	94.81%
Diabetic	217	4.42%	282	5.19%
Age (years)	47.42	(17.17)	47.72	(17.26)
Body mass index (BMI)	25.51	(4.02)	25.56	(5.60)
Health status				
Excellent	1252	25.50%	1155	21.29%
Very good	1213	24.71%	1378	25.40%
Good	1340	27.30%	1598	29.45%
Fair	722	14.71%	948	17.47%
Poor	382	7.78%	347	6.40%
Systolic blood pressure	132.89	(20.99)	129.07	(25.13)

Our final table is much neater and easier to read.

Reference

McDowell, A., A. Engel, J. T. Massey, and K. Maurer. 1981. "Plan and operation of the Second National Health and Nutrition Examination Survey, 1976–1980". In *Vital and Health Statistics*, ser. 1, no. 15. Hyattsville, MD: National Center for Health Statistics.

Also see

[TABLES] **collect recode** — Recode dimension levels in a collection

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