**Example 1** — Table of means, standard deviations, and correlations



# Description

In this example, we demonstrate how to use `table` to compute means and standard deviations, run the `pwcorr` command to obtain a correlation matrix, and create a table with all of these statistics. We also demonstrate how you can use `collect` to customize this table.

# Remarks and examples

Remarks are presented under the following headings:



## Table of correlations

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 explore the correlation between `age`, `weight`, and systolic blood pressure (`bpsystol`).

Let's begin by looking at the correlations with the `pwcorr` command, which allows us to report the significance levels along with the correlation coefficients.

```
. use https://www.stata-press.com/data/r19/nhanes2l
(Second National Health and Nutrition Examination Survey)

. pwcorr age weight bpsystol, sig

             |      age   weight bpsystol
-------------+---------------------------
         age |   1.0000 
             |
             |
      weight |   0.0388   1.0000 
             |   0.0001
             |
    bpsystol |   0.4811   0.2861   1.0000 
             |   0.0000   0.0000
             |
```

pwcorr stores the correlations and significance levels in the matrices r(C) and r(sig), respectively. Below, we create a matrix called vech, which contains the lower triangle elements of the matrix r(C). Then, we list both matrices:

```
. matrix define vech = vech(r(C))

. matrix list r(C)

symmetric r(C)[3,3]
                 age     weight   bpsystol
     age           1
  weight   .03881324          1
bpsystol   .48110277  .28607421          1

. matrix list vech

vech[6,1]
                           c1
          age:age           1
       age:weight   .03881324
     age:bpsystol   .48110277
    weight:weight           1
  weight:bpsystol   .28607421
bpsystol:bpsystol           1
```

We can see that vech() creates a column vector by stacking elements from the first column of r(C), then the second column, and finally the third column. The variable before the colon is the row equation, and the variable after the colon is the row name.

To create our table, we can use collect get to collect the column vectors of both the correlations matrix and the significance levels matrix. We then use collect layout to lay out our table. We place the results for each row name on the rows, by interacting rowname and result, and the row equations (roweq) on the columns:

```
. collect get corr=vech(r(C)) sig=vech(r(sig))

. collect layout (rowname#result) (roweq)

Collection: default
      Rows: rowname#result
   Columns: roweq
   Table 1: 9 x 3
```

| | Age (years) | Weight (kg) | Systolic blood pressure |
|---|---|---|---|
| Age (years) | | | |
| corr | 1 | | |
| sig | . | | |
| Weight (kg) | | | |
| corr | .0388132 | 1 | |
| sig | .0000782 | . | |
| Systolic blood pressure | | | |
| corr | .4811028 | .2860742 | 1 |
| sig | 0 | 3.7e-194 | . |

This is a good starting point, but instead of displaying the significance levels, we would like to display stars representing those levels. We use collect stars to display three stars for values of sig less than 0.01, two stars for values less than 0.05, and one star for values less than 0.1. The attach option tells Stata to attach these stars to the correlations (corr), and shownote adds the note explaining the significance levels that the stars represent.

Additionally, rather than displaying a value of 1 for the correlation of a variable with itself, we want to display a dash. We use `collect style cell` to specify that values of `corr` beyond 0.99 should be labeled with a dash. We also format the correlations to four decimal places and center them horizontally. Whereas before we included all results, now we update our table layout to include only the correlations (`result[corr]`):

```
. collect stars sig 0.01 "***" 0.05 "**" 0.1 "*", attach(corr) shownote

. collect style cell result[corr], maximum(0.99, label("-")) nformat(%6.4f)
> halign(center)

. collect layout (rowname#result[corr]) (roweq)

Collection: default
      Rows: rowname#result[corr]
   Columns: roweq
   Table 1: 6 x 3

------------------------------------------------------------------------
                         | Age (years) Weight (kg) Systolic blood pressure
-------------------------+----------------------------------------------
Age (years)              |
          corr           |      -
Weight (kg)              |
          corr           |  0.0388***        -
Systolic blood pressure  |
          corr           |  0.4811***   0.2861***             -
------------------------------------------------------------------------
*** p<.01, ** p<.05, * p<.1
```

This table looks much better, but we can customize it further. Because we are displaying only correlations, we can hide the labels for the results with `collect style header`. Additionally, we will remove the border on the right side of the corner and row-header sections of the table. The dimension `border_block` divides the table into four sections: `corner`, `column-header`, `row-header`, and `item`. This dimension allows us to modify borders for a whole section of the table. Finally, we add a descriptive title with `collect title` and preview our table with `collect preview`:

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

. collect style cell border_block[corner row-header], border(right, pattern(nil))

. collect title "Table of correlations"

. collect preview

Table of correlations
----------------------------------------------------------------------
                        Age (years) Weight (kg) Systolic blood pressure
----------------------------------------------------------------------
Age (years)                  -
Weight (kg)             0.0388***        -
Systolic blood pressure 0.4811***   0.2861***             -
----------------------------------------------------------------------
*** p<.01, ** p<.05, * p<.1
```

We can now export our table in our preferred style—Word, PDF, HTML, LaTeX, Excel, or Markdown—using `collect export`.

## Table of correlations, means, and standard deviations

Building on our last table, suppose we also want to report the means and standard deviations for each variable. We previously created a column vector with the correlations and significance levels, but for this table, we will create a row vector with those values. The reason is that now we are going to collect other statistics for our table, and the collect system uses the dimensions coleq (column equation) and colname (column name) to identify those results. To align our results, we will need to reference our correlations using column equations and names as well.

To see how this works, let's create another collection for our new table and run our pwcorr command again. We will create a row vector of the correlations by transposing (') the column vector of correlations (vech(r(C))):

```
. collect create corr2
(current collection is corr2)

. pwcorr age weight bpsystol, sig

             |      age   weight bpsystol
-------------+---------------------------
         age |   1.0000 
             |
             |
      weight |   0.0388   1.0000 
             |   0.0001
             |
    bpsystol |   0.4811   0.2861   1.0000 
             |   0.0000   0.0000
             |

. matrix define vechrow = vech(r(C))'

. matrix list r(C)

symmetric r(C)[3,3]
                age     weight   bpsystol
     age          1
  weight  .03881324          1
bpsystol  .48110277  .28607421          1

. matrix list vechrow

vechrow[1,6]
          age:       age:       age:    weight:    weight:  bpsystol:
          age     weight   bpsystol     weight   bpsystol   bpsystol
c1          1  .03881324  .48110277          1  .28607421          1
```

Now, we can identify the correlations by referring to the column equation and column name. We are now ready to create our table.

Previously, we used collect get to collect our results, but with the table command, we can compute summary statistics and incorporate the results from another Stata command in the same table. This versatility will allow us to compute all our statistics with a single table command. With the statistic() option, we request the means and standard deviations of the three variables mentioned. With the command() option, we execute the pwcorr command. We also format the results to display only two digits to the right of the decimal. The arguments before the comma specify how we want to arrange our results. We place the summary statistics (result) and column equations on the rows and the column names of the matrix (colname) on the columns. We also modified the variable label for bpsystol to prevent the table from wrapping.

```
. label variable bpsystol "BP"

. table (result coleq) (colname),
> statistic(mean age weight bpsystol)
> statistic(sd age weight bpsystol)
> command(corr=(vech(r(C))') sig=(vech(r(sig))'):
> pwcorr age weight bpsystol, sig) nformat(%5.2f mean sd)
```

| | Age (years) | Weight (kg) | BP |
|---|---|---|---|
| Mean | | | |
| Mean | 47.58 | 71.90 | 130.88 |
| Standard deviation | | | |
| Standard deviation | 17.21 | 15.36 | 23.33 |
| corr | | | |
| Age (years) | | | |
| pwcorr age weight bpsystol, sig | 1 | .0388132 | .4811028 |
| Weight (kg) | | | |
| pwcorr age weight bpsystol, sig | | 1 | .2860742 |
| BP | | | |
| pwcorr age weight bpsystol, sig | | | 1 |
| sig | | | |
| Age (years) | | | |
| pwcorr age weight bpsystol, sig | . | .0000782 | 0 |
| Weight (kg) | | | |
| pwcorr age weight bpsystol, sig | | . | 3.7e-194 |
| BP | | | |
| pwcorr age weight bpsystol, sig | | | . |

We will use the maximum() option with collect style cell to display a dash for correlations between a variable and itself. And we will display stars for levels of significance, as we did with our previous table. Then, we can lay out our table with the variables (colname) on the rows and the results on the columns. The means and standard deviations can be identified by levels of colname and result. However, the correlations are identified by the value of the column name, column equation, and result. Therefore, we interact coleq with the level corr of result in our table layout.

```
. collect style cell result[corr],
>  maximum(0.99, label("  -  ")) nformat(%6.4f) halign(center)

. collect stars sig 0.01 "***" 0.05 "**" 0.1 "*", attach(corr) shownote

. collect layout (colname) (result[mean sd] coleq#result[corr])

Collection: Table
      Rows: colname
   Columns: result[mean sd] coleq#result[corr]
   Table 1: 3 x 5
```

| | Mean | Standard deviation | Age (years) corr | Weight (kg) corr | BP corr |
|---|---|---|---|---|---|
| Age (years) | 47.58 | 17.21 | - | | |
| Weight (kg) | 71.90 | 15.36 | 0.0388*** | - | |
| BP | 130.88 | 23.33 | 0.4811*** | 0.2861*** | - |

```
*** p<.01, ** p<.05, * p<.1
```

We now have the layout we want, but we will make a few modifications to polish the table further. First, we will hide the label "corr" with collect style header. Then, we will shorten the label for the standard deviations with collect label levels. Rather than repeating the variable labels on the rows and columns, we want to create an index for the variables and use those numbers in the column headers. We can do this by simply modifying the labels for the levels of colname and coleq. Then, we preview our table:

```
. collect style header result[corr], level(hide)
. collect label levels result sd "SD", modify
. collect label levels colname age "1. Age" weight "2. Weight"
> bpsystol "3. Systolic BP", modify
. collect label levels coleq age "1" weight "2" bpsystol "3", modify
. collect preview
```

| | Mean | SD | 1 | 2 | 3 |
|---|---|---|---|---|---|
| 1. Age | 47.58 | 17.21 | - | | |
| 2. Weight | 71.90 | 15.36 | 0.0388*** | - | |
| 3. Systolic BP | 130.88 | 23.33 | 0.4811*** | 0.2861*** | - |

*** p<.01, ** p<.05, * p<.1

Finally, we will center the column headers for the column equations, remove the borders on the right side of the corner and row headers, and provide a title for the table. Both border_block and cell_type divide the table into four sections; the former is used to modify borders, and the latter is used to modify all other appearance styles for the cells. After we make these changes, we preview our table for the last time.

```
. collect style cell cell_type[column-header]#coleq, halign(center)
. collect style cell border_block[corner row-header], border(right, pattern(nil))
. collect title "Descriptive statistics and correlations"
. collect preview
```

Descriptive statistics and correlations

| | Mean | SD | 1 | 2 | 3 |
|---|---|---|---|---|---|
| 1. Age | 47.58 | 17.21 | - | | |
| 2. Weight | 71.90 | 15.36 | 0.0388*** | - | |
| 3. Systolic BP | 130.88 | 23.33 | 0.4811*** | 0.2861*** | - |

*** p<.01, ** p<.05, * p<.1

## 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

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

[TABLES] **collect style column** — Collection styles for column headers

[TABLES] **collect style header** — Collection styles for hiding and showing header components

For suggested citations, see the FAQ on citing Stata documentation.