

## collect stars — Add stars for significant results in a collection

Description

Quick start

Menu

Syntax

Options

Remarks and examples

Stored results

Also see

## Description

`collect stars` manages the creation of stars for indicating the significance of results in a collection. You can also add a note to the table to explain the significance represented by the stars.

## Quick start

Specify three levels of stars based on  $p$ -values in `_r_p` in the `result` dimension

```
collect stars _r_p 0.01 "***" 0.05 "**" 0.1 "*"
```

Same as above, and attach the stars to coefficients stored in `_r_b`

```
collect stars _r_p 0.01 "***" 0.05 "**" 0.1 "*", attach(_r_b)
```

Same as above, and display a note explaining what the stars represent

```
collect stars _r_p 0.01 "***" 0.05 "**" 0.1 "*", attach(_r_b) shownote
```

Clear the current stars specification

```
collect stars, clear
```

## Menu

Statistics > Summaries, tables, and tests > Tables and collections > Build and style table

## Syntax

```
collect stars resultIDs [ #1 "label1"
                        [ #2 "label2"
                        [ #3 "label3"
                        [ #4 "label4"
                        [ #5 "label5" ] ] ] ] [ , options ]
```

*resultIDs* are levels in the `result` dimension whose values determine the stars to be applied.

Value-label pairs are rearranged such that  $\#1 < \#2 < \#3 < \#4 < \#5$ .

For value  $v$  corresponding to one of the results in *resultIDs*,

- if  $v \geq \#5$ , no new stars value is created
- if  $v < \#5$ , the new stars value is set to “*label5*”
- if  $v < \#4$ , the new stars value is set to “*label4*”
- if  $v < \#3$ , the new stars value is set to “*label3*”
- if  $v < \#2$ , the new stars value is set to “*label2*”
- if  $v < \#1$ , the new stars value is set to “*label1*”

<i>options</i>	Description
<b>Main</b>	
<code>name(<i>cname</i>)</code>	specify stars for collection <i>cname</i>
<code>clear</code>	remove previous style properties
<code>attach(<i>attachres</i>)</code>	specify that stars be appended to items in <i>attachres</i>
<code>fortags(<i>taglist</i>)</code>	specify tags identifying items for which to generate new star items
<code>result</code>	tag new items with <code>result[stars]</code>
<code>dimension</code>	tag new items using <code>stars[label]</code>
<b>Options</b>	
<code>[no]shownote</code>	specify whether to display the stars note
<code>increasing</code>	compose stars note with increasing <i>p</i> -values; the default
<code>decreasing</code>	compose stars note with decreasing <i>p</i> -values
<code>pvname(<i>string</i>)</code>	specify a name for <i>p</i> -value in the stars note
<code>delimiter(<i>string</i>)</code>	specify label delimiter in the stars note
<code>nformat(<i>%fmt</i>)</code>	specify numeric format for cutoff values in the stars note
<code>prefix(<i>string</i>)</code>	specify a prefix for the stars note
<code>suffix(<i>string</i>)</code>	specify a suffix for the stars note

## Options

### Main

`name` (*cname*) specifies the collection to which stars are to be applied. By default, stars are applied to the current collection.

`clear` removes existing `collect stars` properties.

`attach` (*attachres*) specifies that items with result levels in *attachres* be appended with the associated star label when rendered to a table.

`fortags` (*taglist*) specifies conditions for selecting which values are to be used to create stars. Values with tags in *taglist* will be used to generate new stars.

Within the *taglist*, if tags are joined by #, values having all of these tags are selected; if tags are separated by a space, values with any of these tags are selected.

*taglist* contains

*tagspec*

*tagspec taglist*

*tagspec* contains

*tag*

*tag#tag[#tag[. . .]]*

*tag* contains

*dimension*

*dimension [levels]*

*dimension* is a dimension in the collection.

*levels* are levels of the corresponding dimension.

Distinguish between `[]`, which are to be typed, and `[][]`, which indicate optional arguments.

`result` and `dimension` control how `collect stars` adds items when labeling significant results. These options are mutually exclusive.

`result` specifies the factory default behavior, and this option is necessary only if the following `dimension` behavior is in effect and you want to change back to the `result` behavior.

`dimension` specifies that `dimension stars` be added to the collection. Items will be tagged with `stars[value]`, and the labels will be tagged with `stars[label]`. Use this option for layouts where results are to be stacked within columns, and use `new dimension stars` in the column specification of the layout.

### Options

`noshownote` and `shownote` control whether to show the stars note.

`noshownote` suppresses the stars note. This is the default.

`shownote` specifies to show the stars note.

increasing and decreasing control the order of  $p$ -values in the stars note.

increasing specifies that the stars note be composed with increasing  $p$ -values. This is the default.

decreasing specifies that the stars note be composed with decreasing  $p$ -values.

pvname(*string*) specifies a name for  $p$ -value in the stars note. The default is pvname(p).

delimiter(*string*) specifies the delimiter between labels in the stars note. The default is delimiter(", ").

nformat(*%fmt*) specifies the numeric format for the cutoff values in the stars note. The default is nformat(%9.0g).

prefix(*string*) specifies the prefix for the stars note. The prefix is empty by default.

suffix(*string*) specifies the suffix for the stars note. The suffix is empty by default.

## Remarks and examples

Stars are often used in tables to denote significance. `collect stars` allows you to include stars in your table based on other values, typically  $p$ -values that are already in your collection. You can attach the stars to a statistic, typically coefficients, or present them separately. Whichever style you choose, you will likely want to use the `shownote` option to add a note to your table explaining the significance level that the stars represent.

`collect stars` stores its specification among the collection's style properties. This means that the creation of stars for significant results can be made before collecting results, can be saved to disk via `collect style save`, and can be applied to other collections via `collect style use`. Additionally, you can check the current stars specification with `collect query stars`.

To demonstrate, we first create a table of regression results displaying coefficients and  $p$ -values.

```
. use https://www.stata-press.com/data/r19/nhanes2
. quietly collect: regress bpsystol bmi i.region age
. collect style showbase off
. collect layout (colname) (result[_r_b _r_p])
Collection: default
  Rows: colname
  Columns: result[_r_b _r_p]
Table 1: 6 x 2
```

	Coefficient p-value	
Body mass index (BMI)	1.303719	0.000
MW	-.0659707	0.907
S	-.5170085	0.356
W	-.6045511	0.289
Age (years)	.5887217	0.000
Intercept	69.89029	0.000

Rather than showing the  $p$ -values, we can use `collect stars` to define the levels of the  $p$ -values stored in `_r_p` for which stars should be shown. Here we will use three stars for values less than 0.01, two stars for values less than 0.05, and one star for values less than 0.1. A new `stars` level in the `result` dimension is created and can be used in our table layout.

```
. collect stars _r_p 0.01 "****" 0.05 "***" 0.1 "*"
. collect layout (colname) (result[_r_b stars])
Collection: default
  Rows: colname
  Columns: result[_r_b stars]
Table 1: 6 x 2
```

	Coefficient stars
Body mass index (BMI)	1.303719 ***
MW	-.0659707
S	-.5170085
W	-.6045511
Age (years)	.5887217 ***
Intercept	69.89029 ***

It is unlikely that we want the level name stars to show in the column header. It would also be helpful to left-align the stars to be closer to the reported coefficients. We can do this with `collect style header` and `collect style cell`.

```
. collect style header result[stars], level(hide)
. collect style cell result[stars], halign(left)
. collect preview
```

	Coefficient
Body mass index (BMI)	1.303719 ***
MW	-.0659707
S	-.5170085
W	-.6045511
Age (years)	.5887217 ***
Intercept	69.89029 ***

Alternatively, we can directly attach the stars to the coefficient by specifying the `attach()` option and naming the result (`_r_b`) that we want the stars attached to.

```
. collect stars _r_p 0.01 "****" 0.05 "***" 0.1 "*" "1 " " ", attach(_r_b)
. collect layout (colname) (result[_r_b])
Collection: default
  Rows: colname
  Columns: result[_r_b]
Table 1: 6 x 1
```

	Coefficient
Body mass index (BMI)	1.303719***
MW	-.0659707
S	-.5170085
W	-.6045511
Age (years)	.5887217***
Intercept	69.89029***

Here we added extra spaces to force three characters following the number in each cell. This gives nice alignment when we are looking at the results in formats such as plain text and the Stata Markup and Control Language format. However, if you are exporting your table to other formats, you may prefer to leave the stars in a separate column and apply alignment and margin styles to achieve your desired look.

It is common to want to stack coefficients and their standard errors in a single column of an estimation table. While the default collect stars behavior tends to yield ugly tables for this layout, the solution is to use option dimension and put the new stars dimension in the column specification of the layout.

```
. collect stars _r_p 0.01 "****" 0.05 "***" 0.1 "*", attach(_r_b) dimension
. collect layout (colname#result[_r_b _r_se]) (stars)
Collection: default
  Rows: colname#result[_r_b _r_se]
  Columns: stars
  Table 1: 18 x 2
```

Body mass index (BMI)	
Coefficient	1.303719 ***
Std. error	.0395032
MW	
Coefficient	-.0659707
Std. error	.5633352
S	
Coefficient	-.5170085
Std. error	.559805
W	
Coefficient	-.6045511
Std. error	.5698782
Age (years)	
Coefficient	.5887217 ***
Std. error	.0112852
Intercept	
Coefficient	69.89029 ***
Std. error	1.142773

Although the stars we have chosen to represent the  $p$ -values may be common in practice, to be clear, we add an explanatory note below. Note that we do not have to respecify our labels; we are simply adding on to our current stars specification:

```
. collect stars, shownote
. collect preview
```

Body mass index (BMI)		
Coefficient	1.303719	***
Std. error	.0395032	
MW		
Coefficient	-.0659707	
Std. error	.5633352	
S		
Coefficient	-.5170085	
Std. error	.559805	
W		
Coefficient	-.6045511	
Std. error	.5698782	
Age (years)		
Coefficient	.5887217	***
Std. error	.0112852	
Intercept		
Coefficient	69.89029	***
Std. error	1.142773	

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

## Stored results

collect stars stores the following in `s()`:

Macros

<code>s(collection)</code>	name of collection
<code>s(value1)</code>	star cutoff value 1
<code>s(label1)</code>	star label 1
<code>s(value2)</code>	star cutoff value 2
<code>s(label2)</code>	star label 2
<code>s(value3)</code>	star cutoff value 3
<code>s(label3)</code>	star label 3
<code>s(value4)</code>	star cutoff value 4
<code>s(label4)</code>	star label 4
<code>s(value5)</code>	star cutoff value 5
<code>s(label5)</code>	star label 5

## Also see

[TABLES] [collect clear](#) — Clear all collections in memory

[TABLES] [collect drop](#) — Drop collections from memory

[TABLES] [collect query](#) — Query collection style properties

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on [citing Stata documentation](#).

