Title

stata.com

collect remap — Remap tags in a collection

Description

`collect remap` remaps tags associated with values in a collection. Remapping tags can be useful when you need to specify a table layout but the original tags do not allow you to place values from different commands that are tagged differently into the same rows, columns, or tables.

With `collect remap`, you can remap tags for levels of an existing dimension to a new dimension with the same levels, remap tags for levels of an existing dimension to a new dimension with new levels, or remap tags for levels of an existing dimension to new levels within the existing dimension.

Quick start

Remap all tags with dimension `olddim` to new dimension `newdim`, with the level unchanged

```
collect remap olddim = newdim
```

Remap tags with levels `lev1` and `lev2` in dimension `olddim` to `newdim`, with the level unchanged

```
collect remap olddim[lev1 lev2] = newdim
```

As above, but remap tags to the specified levels of the new dimension

```
collect remap olddim[lev1 lev2] = newdim[lev4 lev3]
```

Menu

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

Syntax

Remap tags from an existing dimension to a new dimension, with the level unchanged

```
collect remap olddim = newdim [ , name(cname) fortags(taglist) ]
```

Remap tags with specified levels of an existing dimension to a new dimension, with the level unchanged

```
collect remap olddim[oldlevels] = newdim [ , name(cname) fortags(taglist) ]
```

Remap tags with specified levels of an existing dimension to new levels of a new dimension

```
collect remap olddim[oldlevels] = newdim[newlevels]
    [ , name(cname) fortags(taglist) ]
```

where *olddim* is the name of an existing dimension in the collection, *newdim* is the name of a dimension into which levels of *oldim* are to be mapped, *oldlevels* are the names of existing levels in the dimension, and *newlevels* are the names of the levels to which *oldlevels* are to be set.

Levels `_r_ci` and `_r_cri` of dimension `result` are not allowed in *oldlevels*.

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

Options

Main

`name(`*cname*`)` specifies the collection in which to remap items. If this option is not specified, the change is made in the current collection.

Options

`fortags(`*taglist*`)` specifies conditions for selecting the values to which remapped tags will be applied. Values with tags in *taglist* will have their tags remapped.

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.

Levels `_r_ci` and `_r_cri` of dimension `result` are not allowed in *taglist*.

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

Remarks and examples

stata.com

After collecting results, we occasionally need to remap tags to lay out the table that we wish to create. `collect remap` allows you to remap tags from the existing levels of an existing dimension to new tags, possibly with new dimensions and new levels.

To demonstrate, we use data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981). With the `table` command, we create a table with two regression results as well as the means for each dependent variable.

```
. use https://www.stata-press.com/data/r17/nhanes2
. quietly table (result colname) (statcmd),
>     command(regress bpsystol age weight)
>     command(regress bpdiast age weight)
>     statistic(mean bpsystol bpdiast) nformat(%6.3f)
. collect style header statcmd, level(value)
. collect preview
```

	1	2	3
Mean			
Systolic blood pressure			130.882
Diastolic blood pressure			81.715
Coefficient			
Age (years)	0.638	0.188	
Weight (kg)	0.407	0.312	
Intercept	71.271	50.376	

The `statcmd` dimension is used to identify the columns of the table. The regression results are tagged with `statcmd[1]` and `statcmd[2]` for `bpsystol` and `bpdiast`, respectively. The means of the dependent variables are tagged with `statcmd[3]`. We can use `collect remap` to remap the `statcmd[3]` tags so that the mean of each dependent variable has the same level as the corresponding regression results.

```
. collect remap statcmd[3]=statcmd[1], fortags(var[bpsystol])
(1 items remapped in collection Table)
. collect remap statcmd[3]=statcmd[2], fortags(var[bpdiast])
(1 items remapped in collection Table)
. collect preview
```

	1	2
Mean		
Systolic blood pressure	130.882	
Diastolic blood pressure		81.715
Coefficient		
Age (years)	0.638	0.188
Weight (kg)	0.407	0.312
Intercept	71.271	50.376

Because we wanted to remap only `statcmd[3]` to `statcmd[1]` for the mean value of `bpsystol`, we specify `fortags(var[bpsystol])`, which indicates that the remapping will be performed only for values with this tag. Likewise, we remap `statcmd[3]` to `statcmd[2]` only for values with the tag `var[bpdiast]`. This produced a table with only two columns, one for each dependent variable.

Our rows are identified by the `result` and `colname` dimensions. Because our means have different levels of `colname`, they appear on separate rows. We can place them on the same row by remapping the separate `bpsystol` and `bpdiast` levels to one level, say, `mean`.

```
. collect remap colname[bpsystol bpdiast] = colname[mean mean]
(2 items remapped in collection Table)
. collect preview
```

	1	2
Mean		
mean	130.882	81.715
Coefficient		
Age (years)	0.638	0.188
Weight (kg)	0.407	0.312
Intercept	71.271	50.376

Now, we have the values arranged where we would like them in our table. We can clean up the row and column headers of our table by typing

```
. collect label levels statcmd 1 "Systolic BP" 2 "Diastolic BP", modify
. collect style header statcmd, level(label)
. collect label levels result mean "Mean of dependent variable"
>    _r_b "Coefficients", modify
. collect style header colname[mean], level(hide)
. collect preview
```

	Systolic BP	Diastolic BP
Mean of dependent variable	130.882	81.715
Coefficients		
Age (years)	0.638	0.188
Weight (kg)	0.407	0.312
Intercept	71.271	50.376

See [TABLES] **collect label** and [TABLES] **collect style header** for more information on these commands.

In the examples above, we remapped tags to new levels within the same dimension. We could have performed these same remappings using `collect recode`. However, `collect remap` can do more. We could, for instance, type

```
. collect remap colname[bpsystol bpdiast] = mycol
```

to remap the existing tags to tags with new dimension `mycol` but with the existing level names. We could also type

```
. collect remap colname[bpsystol bpdiast] = mycol[mean mean]
```

and remap the existing tags to tags with new dimension `mycol` and level `mean`.

Stored results

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

Macros

`s(collection)`	name of collection
`s(k_remapped)`	number of remapped items

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. *Vital and Health Statistics* 1(15): 1–144.

Also see

[TABLES] **collect addtags** — Add tags to items in a collection

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