**collect recode** — Recode dimension levels in a collection

---

**Description**

`collect recode` recodes dimension levels attached to values in the current collection.

**Quick start**

Recode level `lev1` to `newlevel` and level `lev2` to `newlevel2` in dimension `dim1`

```
collect recode dim1 lev1=newlevel  lev2=newlevel2
```

As above, but apply the recoded levels only to values in the collection tagged with `dim2[lev3]`

```
collect recode dim1 lev1=newlevel  lev2=newlevel2, fortags(dim2[lev3])
```

Recode levels `2.catvar` and `3.catvar` in dimension `dim2`

```
collect recode dim2 2.catvar=catvar2 3.catvar=catvar3
```

---

**Menu**

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

---

**Syntax**

```
collect recode  dim  oldlevel = newlevel  [oldlevel = newlevel  ...]
[ ,  name(cname)  fortags(taglist)]
```

where `dim` is the name of a dimension in the collection, `oldlevel` is the name of an existing level in the dimension, and `newlevel` is the name of the level to which `oldlevel` is to be set.

Levels `r_ci` and `r_cri` of dimension `result` are not allowed in `oldlevel`.

---

**Options**

- `name(cname)` specifies the collection in which to recode the levels of the dimension. 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 the recoded levels will be applied. Values with tags in taglist will have their levels recoded.

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 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

After collecting results, we occasionally need to recode levels of a dimension to lay out the table that we wish to create. collect recode replaces the existing levels of a dimension with newly specified 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

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Systolic blood pressure</td>
<td>130.882</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diastolic blood pressure</td>
<td>81.715</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coefficient</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age (years)</td>
<td>0.638</td>
<td>0.188</td>
<td></td>
</tr>
<tr>
<td>Weight (kg)</td>
<td>0.407</td>
<td>0.312</td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>71.271</td>
<td>50.376</td>
<td></td>
</tr>
</tbody>
</table>
```
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 recode` to recode the levels of `statcmd` so that the mean of each dependent variable has the same level as the corresponding regression results.

```plaintext
. collect recode statcmd 3 = 1, for tags(var[bpsystol])
(1 items recoded in collection Table)
. collect recode statcmd 3 = 2, for tags(var[bpdiast])
(1 items recoded in collection Table)
. collect preview

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Systolic blood pressure</td>
<td>130.882</td>
<td></td>
</tr>
<tr>
<td>Diastolic blood pressure</td>
<td></td>
<td>81.715</td>
</tr>
<tr>
<td><strong>Coefficient</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age (years)</td>
<td>0.638</td>
<td>0.188</td>
</tr>
<tr>
<td>Weight (kg)</td>
<td>0.407</td>
<td>0.312</td>
</tr>
<tr>
<td>Intercept</td>
<td>71.271</td>
<td>50.376</td>
</tr>
</tbody>
</table>
```

Because we wanted to recode only `statcmd[3]` to `statcmd[1]` for the mean value of `bpsystol`, we specify `for tags(var[bpsystol])`, which indicates that the recode will be performed only for values with this tag. Likewise, we recode `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 recoding the separate `bpsystol` and `bpdiast` levels to one level, say, `mean`.

```plaintext
. collect recode colname bpsystol = mean
(1 items recoded in collection Table)
. collect recode colname bpdiast = mean
(1 items recoded in collection Table)
. collect preview

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>130.882</td>
<td>81.715</td>
</tr>
<tr>
<td><strong>Coefficient</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age (years)</td>
<td>0.638</td>
<td>0.188</td>
</tr>
<tr>
<td>Weight (kg)</td>
<td>0.407</td>
<td>0.312</td>
</tr>
<tr>
<td>Intercept</td>
<td>71.271</td>
<td>50.376</td>
</tr>
</tbody>
</table>
```

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

```plaintext
. 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

<table>
<thead>
<tr>
<th>Systolic BP</th>
<th>Diastolic BP</th>
</tr>
</thead>
<tbody>
<tr>
<td>130.882</td>
<td>81.715</td>
</tr>
</tbody>
</table>

Coefficients

<table>
<thead>
<tr>
<th>Systolic BP</th>
<th>Diastolic BP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.638</td>
<td>0.188</td>
</tr>
<tr>
<td>0.407</td>
<td>0.312</td>
</tr>
<tr>
<td>71.271</td>
<td>50.376</td>
</tr>
</tbody>
</table>

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

### Stored results

`collect recode` stores the following in $s()$:

<table>
<thead>
<tr>
<th>Macros</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(collection)</td>
</tr>
<tr>
<td>s(dimname)</td>
</tr>
<tr>
<td>s(k_recoded)</td>
</tr>
</tbody>
</table>

### Reference


### Also see

- [TABLES] `collect addtags` — Add tags to items in a collection
- [TABLES] `collect remap` — Remap tags in a collection