

matrix rownames — Name rows and columns[Description](#)[Syntax](#)[Remarks and examples](#)[Also see](#)

Description

`matrix rownames` and `colnames` reset the row and column names of an already existing matrix.

`matrix roweq` and `coleq` also reset the row and column names of an already existing matrix, but if a simple name (a name without a colon) is specified, it is interpreted as an equation name.

In either case, the part of the name not specified is left unchanged.

Syntax

Reset row names of matrix

```
matrix rownames A = names
```

Reset column names of matrix

```
matrix colnames A = names
```

Reset row names and interpret simple names as equation names

```
matrix roweq A = names
```

Reset column names and interpret simple names as equation names

```
matrix coleq A = names
```

where *names* can be

- a simple name;
- an interaction;
- a colon followed by a simple name;
- a colon followed by an interaction;
- an equation name followed by a colon;
- an equation name, a colon, and a simple name; or
- an equation name, a colon, and an interaction.

A simple name is a sequence of 1 to 32 characters, which can include, for example, digits, spaces, and any Unicode letter; colons should not be used. If the name includes a space, it must be enclosed in double quotes. A simple name may be augmented with time-series operators and factor-variable specifications; these operators do not contribute to the 32-character limit.

An interaction is two or more simple names delimited by #. Spaces are not allowed in interactions.

An equation name is a sequence of 1 to 32 characters, which can include, for example, digits, spaces, and any Unicode letter; colons should not be used. If the name includes a space, it must be enclosed in double quotes.

When specifying both an equation name and a simple name, the entire string must be enclosed in quotes if either the equation name or the simple name contains spaces.

Remarks and examples

[stata.com](http://www.stata.com)

See [U] [14.2 Row and column names](#) for a description of the row and column names bordering a matrix.

▷ Example 1

In general, the names bordering matrices are set correctly by Stata because of the tracking of the matrix algebra, and you will not need to reset them. Nevertheless, imagine that you have formed $X'X$ in the matrix named `XX` and that it corresponds to the underlying variables `price`, `weight`, and `mpg`:

```
. matrix list XX
symmetric XX[3,3]
      c1      c2      c3
r1  3.448e+09
r2  1.468e+09  7.188e+08
r3   9132716  4493720   36008
```

You did not form this matrix with `matrix accum` because, had you done so, the rows and columns would already be correctly named. However you formed it, you now want to reset the names:

```
. matrix rownames XX = price weight mpg
. matrix colnames XX = price weight mpg
. matrix list XX
symmetric XX[3,3]
      price      weight      mpg
price  3.448e+09
weight 1.468e+09  7.188e+08
mpg    9132716  4493720   36008
```

◀

▷ Example 2

We now demonstrate setting the equation names and names with time-series operators.

```
. matrix list AA
symmetric AA[4,4]
      c1      c2      c3      c4
r1   .2967663
r2   .03682017  .57644416
r3  -.87052852  .32713601  20.274957
r4  -1.572579  -.63830843  -12.150097  26.099582
. matrix rownames AA = length L3D2.length mpg L.mpg
. matrix colnames AA = length L3D2.length mpg L.mpg
. matrix roweq AA = eq1 eq1 eq2 eq2
. matrix coleq AA = eq1 eq1 eq2 eq2
```

```
. matrix list AA
symmetric AA[4,4]
           eq1:      eq1:      eq2:      eq2:
           L3D2.      L3D2.      L.
           length    length    mpg      mpg
eq1:length .2967663
eq1:L3D2.length .03682017 .57644416
eq2:mpg -.87052852 .32713601 20.274957
eq2:L.mpg -1.572579 -.63830843 -12.150097 26.099582
```

Factor variables and interactions are much like time-series-operated variables, we specify each level variable.

```
. mat rownames AA = 0b.foreign 1.foreign 0.foreign#c.mpg 1.foreign#c.mpg
. mat colnames AA = 0b.foreign 1.foreign 0.foreign#c.mpg 1.foreign#c.mpg
```

As in factor-variable varlists, we can combine any time-series lead and lag operators with factor variables.

```
. mat rownames XX = 0bL2.foreign 1L2.foreign 0L3.foreign#cL3.mpg
> 1L3.foreign#cL3.mpg
. mat colnames XX = 0bL2.foreign 1L2.foreign 0L3.foreign#cL3.mpg
> 1L3.foreign#cL3.mpg
```

◀

□ Technical note

`matrix rownames` and `colnames` sometimes behave in surprising ways:

1. If your list of names includes no colons—does not mention the equation names—whatever equation names are in place are left in place; they are not changed.
2. If your list of names has every name ending in a colon—so that it mentions only the equation names and not the subnames—whatever subnames are in place are left in place; they are not changed.
3. If your list of names has fewer names than are required to label all the rows or columns, the last name in the list is replicated. (If you specify too many names, you will get the conformability error message, and no names will be changed.)

These surprises have their uses, but if you make a mistake, the result really may surprise you. For instance, rule 3, by itself, is just odd. Combined with rule 2, however, rule 3 allows you to set all the equation names in a matrix easily. If you type `'matrix rownames XX = myeq:'`, all the equation names in the row are reset while the subnames are left unchanged:

```
. matrix rownames XX = myeq:
. matrix list XX
symmetric XX[3,3]
           price      weight      mpg
myeq:price 3.448e+09
myeq:weight 1.468e+09 7.188e+08
myeq:mpg 9132716 4493720 36008
```

Setting equation names is often done before forming a partitioned matrix so that, when the components are assembled, each has the correct equation name.

Thus to review, to get the result above, we could have typed

```
. matrix rownames XX = myeq:price myeq:weight myeq:mpg
```

or

```
. matrix rownames XX = price weight mpg
. matrix rownames XX = myeq:
```

or even

```
. matrix rownames XX = myeq:
. matrix rownames XX = price weight mpg
```

All would have resulted in the same outcome. The real surprise comes, however, when you make a mistake:

```
. matrix rownames XX = myeq:
. matrix rownames XX = price weight
. matrix list XX
symmetric XX[3,3]
      price      weight      mpg
myeq:price 3.448e+09
myeq:weight 1.468e+09 7.188e+08
myeq:weight 9132716 4493720 36008
```

Our mistake above is that we listed only two names for the subnames of the rows of `XX` and `matrix rownames` and then labeled both of the last rows with the subtitle `weight`.

□

□ Technical note

The equation name `_:` by itself is special; it means the null equation name. For instance, as of the last technical note, we were left with

```
. matrix list XX
symmetric XX[3,3]
      price      weight      mpg
myeq:price 3.448e+09
myeq:weight 1.468e+09 7.188e+08
myeq:weight 9132716 4493720 36008
```

Let's fix it:

```
. matrix rownames XX = price weight mpg
. matrix rownames XX = _:
. matrix list XX
symmetric XX[3,3]
      price      weight      mpg
price 3.448e+09
weight 1.468e+09 7.188e+08
mpg 9132716 4493720 36008
```

□

□ Technical note

`matrix roweq` and `matrix coleq` are really the same commands as `matrix rownames` and `matrix colnames`. They differ in only one respect: if a specified name does not contain a colon, `matrix roweq` and `matrix coleq` interpret that name as if it did end in a colon.

`matrix rownames`, `matrix colnames`, `matrix roweq`, and `matrix coleq` are often used in conjunction with the `rowfullnames`, `colfullnames`, `rownames`, `colnames`, `roweq`, and `coleq` macro functions introduced in [P] [matrix define](#). The `rownames` and `colnames` macro functions return only the name, including any time-series or factor-variable operators, but not the equation name.

```
. matrix list AA
symmetric AA[4,4]
           eq1:      eq1:      eq2:      eq2:
           L3D2.    L3D2.    L.
           length   length   mpg       mpg
eq1:length   .2967663
eq1:L3D2.length .03682017 .57644416
eq2:mpg     -1.87052852 .32713601 20.274957
eq2:L.mpg   -1.572579  -.63830843 -12.150097 26.099582

. local rsubs : rownames AA
. display "The row subnames of AA are -- 'rsubs' --"
The row subnames of AA are -- length L3D2.length mpg L.mpg --
```

Similarly, the `roweq` macro function returns only the equation names without the trailing colon:

```
. local reqs : roweq AA
. display "The row equations of AA are -- 'reqs' --"
The row equations of AA are -- eq1 eq1 eq2 eq2 --
```

Now consider the problem that you have two matrices named A and B that have the same number of rows. A is correctly labeled and includes equation names. You want to copy the complete names of A to B. You might be tempted to type

```
. local names : rownames A
. matrix rownames B = 'names'
```

This is not adequate. You will have copied the names but not the equation names. To copy both parts of the complete names, you can type

```
. local subs : rownames A
. local eqs : roweq A
. matrix rownames B = 'subs'
. matrix roweq B = 'eqs'
```

This method can be used even when there might not be equation names. The equation name `_` is special for two reasons: setting an equation to that name removes the equation name, and when there is no equation name, the `roweq` and `coleq` macro functions return that name.

A better way to copy the names is to use the `rowfullnames` and `colfullnames` macro functions (see [P] [matrix define](#) and [P] [macro](#)). You can more compactly type

```
. local rname : rowfullnames A
. matrix rownames B = 'rname'
```

Also see

- [P] [macro](#) — Macro definition and manipulation
- [P] [matrix](#) — Introduction to matrix commands
- [P] [matrix define](#) — Matrix definition, operators, and functions
- [U] [14 Matrix expressions](#)

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. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.

