spmatrix matafromsp — Copy weighting matrix to Mata

Description

`spmatrix matafromsp` copies weighting matrix *spmatname* from Sp to Mata. Weighting matrix *spmatname* remains unchanged.

Quick start

Create weighting matrix `W` and ID vector `id` in Mata from spatial weighting matrix `C`

```
spmatrix matafromsp W id = C
```

Menu

Statistics > Spatial autoregressive models

Syntax

`spmatrix matafromsp` *matamatrix matavec* `=` *spmatname*

Remarks and examples

Remarks are presented under the following headings:

Getting W and id

The command

```
. spmatrix matafromsp W id = C
```

copies spatial weighting matrix `C` to a Mata matrix named `W` and copies `C`'s `_ID` vector to a Mata vector named `id`.

What is id? When a spatial weighting matrix such as C is created, stored along with it are the _ID values. Those _ID values identify the meaning of the rows and columns.

Consider a spatial weighting matrix created by spmatrix create. We use the datasets downloaded in [SP] **estat moran**.

```
. use homicide1990
(S.Messner et al.(2000), U.S southern county homicide rates in 1990)
. spset
  (output omitted)
. spmatrix create contiguity C
```

What is the meaning of element $c_{1,2}$? It is the spillover from _ID[2] to _ID[1]. If the data were currently spset on fips, _ID[1] might equal 48507 and _ID[2] might equal 48003, and thus it would be the spillover from Andrews to Zavala county in Texas. Sp keeps a copy of the _ID vector so that later, when the data are in a different order, $c\{1,2\}$ will still mean the spillover from Andrews to Zavala county.

You need not concern yourself with id if you plan on doing something with W that does not involve the data in memory. If what you need to do involves the data in memory, you will need to address the problem that the order of the data in memory now is not the same as it was when W was created.

Using W without involving the data in memory

Say that you wish to fetch C from Sp just so you can change values greater than or equal to 0.8 to 0.5. Doing that does not involve the data in memory. You type

```
. spmatrix matafromsp W id = C
. mata:
------------------------------------------------- mata (type end to exit) -----
: for (i=1; i<=rows(W); i++) {
>         for (j=1; j<=cols(W); j++) {
>                 if (W[i,j] >= 0.8) W[i,j] = 0.5
>         }
> }
: end
-------------------------------------------------------------------------------
```

You might now store the W back into C by typing

```
. spmatrix spfrommata C = W id, replace
```

You specify the same id vector you received because you have not changed the ordering of the rows or columns of the matrix.

Using W involving the data in memory

If you intend to use W and the data in memory together, you need to align the data and W. The instructions presented here work with cross-sectional data but not panel data.

First, check whether the data and W are conformable:

```
. mata:
------------------------------------------------- mata (type end to exit) -----
: ID = st_data(., "_ID")
: assert( sort(ID, 1) == sort(id, 1) )
: end
-------------------------------------------------------------------------------
```

If Mata reports that the assertion is false, then the data and W are not conformable. This has nothing to do with observations and rows and columns being in different order. Not conformable means that one, the other, or both are missing _ID values that the other one has.

Let's imagine that Mata responds with silence to the assertion. Thus, the data are conformable. If they are also in the same order, you can use the data and W together, so find out if they are.

```
. mata:
------------------------------------------------- mata (type end to exit) -----
: assert( ID == id )
: end
-------------------------------------------------------------------------------
```

If they are in the same order, row/column 1 of the matrix corresponds to observation 1 of the data, row/column 2 of the matrix corresponds to observation 2 of the data, and so on.

If Mata reports that the assertion is false, you have to put the data in the same order. Here is how:

```
. mata:
------------------------------------------------- mata (type end to exit) -----
: p  = order(id, 1)
: W  =  W[p, p]         // put W in ascending order of id
: id = id[p]            // put id in ascending order of id
: end
-------------------------------------------------------------------------------

. sort _ID              // put the data in ascending order of _ID
```

You can now do whatever with W and the data. Row/column 1 of W corresponds to observation 1 of the data, row/column 2 of W corresponds to observation 2 of the data, and so on.

Perhaps whatever you will do involves, as a last step, posting the matrix back to Sp. In that case, use the id variable you updated:

```
. spmatrix spfrommata C = W id, replace
```

Also see

[SP] **spmatrix** — Categorical guide to the spmatrix command

[SP] **spmatrix spfrommata** — Copy Mata matrix to Sp

[SP] **Intro** — Introduction to spatial data and SAR models

Mata Reference Manual