spmatrix spfrommata — Copy Mata matrix to Sp

Description Quick start Menu Syntax
Options Remarks and examples References Also see

Description

`spmatrix spfrommata` copies a weighting matrix and an ID vector from Mata to an Sp spatial weighting matrix.

Quick start

Create Sp spatial weighting matrix `Wnew` from Mata matrix `W` and vector `v` with the default spectral normalization

```
spmatrix spfrommata Wnew = W v
```

Menu

Statistics > Spatial autoregressive models

Syntax

```
spmatrix spfrommata spmatname = matamatrix matavec [ , options ]
```

options	Description
`normalize(`*normalize*`)`	type of normalization; default is `normalize(spectral)`
`replace`	replace existing weighting matrix

Options

`normalize(`*normalize*`)` specifies how the resulting matrix is to be scaled. `normalize(spectral)` is the default. `normalize(minmax)`, `normalize(row)`, and `normalize(none)` are also allowed. See [SP] **spmatrix create** for full details of the option and *Choosing weighting matrices and their normalization* in [SP] **spregress** for details about normalization.

`replace` specifies that matrix *spmatname* be overwritten if it already exists.

Remarks and examples

Remarks are presented under the following headings:

W and v
Simple use
Advanced use

W and v

Two components are required to set an Sp spatial weighting matrix: the spatial weighting matrix itself and its vector of _ID values. Let's call them W and v, respectively. v states that the first row and column of W correspond to _ID==v[1], the second row and column correspond to _ID==v[2], and so on. The purpose of v and how it works is explained in [SP] **spmatrix matafromsp**.

Examples of spmatrix spfrommata can be found in [SP] **spmatrix create** and [SP] **spmatrix matafromsp**.

Simple use

We are going to show you how Mata can be used to construct complicated spatial weighting matrices. However, we will start with a simple case in which the values of the weighting matrix $W_{i,j}$ are a function of variables in observations i and j of the data in memory. Inverse-distance matrices are an example of this. The distance between i and j is a function of the values of Stata variables _CX and _CY in observations i and j.

We start by loading the Sp data into memory:

```
. use your_sp_data
```

The Mata solution is

```
. mata:
------------------------------------------------- mata (type end to exit) -----
: id = st_data(., "_ID")
: location = st_data(., ("_CX", "_CY"))
: N = st_nobs()
: W = J(N, N, 0)
: for (i=1; i<=N; i++) {
>         for (j=1; j<i; j++) {
>                 delta = location[i,.] - location[j,.]
>                 W[i,j] = W[j,i] = 1/sqrt(delta*delta')
>         }
> }
: end
--------------------------------------------------------------------------------

. spmatrix spfrommata myIdist = W id
```

We just created an inverse-distance matrix. If you wanted to create such matrices, it would obviously be easier to type

```
. spmatrix create idistance myIdist
```

The Mata solution has the advantage that you can substitute different inverse-distance functions, such as inverse-distance squared. There is an easier solution for that case too, namely, the one outlined in [SP] **spmatrix userdefined**.

Now that you know how Mata and spmatrix spfrommata work in simple cases, we can show you an example that could be done no other way than by direct use of Mata.

Advanced use

You have `export.dta`, a cross-sectional Sp dataset on countries and their characteristics. Its `_ID` variable contains standard country codes. You need to construct a spatial weighting matrix to use with it.

`export.dta` does not itself contain sufficient information to construct the matrix you want to use. Instead, you have a second dataset, which is not Sp. It contains

```
. copy https://www.stata-press.com/data/r19/exports.dta .
. use exports
(Country export data)
. describe
Contains data from https://www.stata-press.com/data/r19/exports.dta
 Observations:        38,220                  Country export data
    Variables:             4                  17 Apr 2024 08:26
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
from            int     %9.0g                 From (country code)
to              int     %9.0g                 To (country code)
exports         float   %9.0g                 Exports ($ value)
gdp             float   %9.0g                 GDP of producer
-------------------------------------------------------------------------------
Sorted by:
```

The data record exports from one country to another for all 196 countries of the world. We say exports, but we can just as well interpret the data as imports by reversing the roles of variables `from` and `to`.

To simplify the problem, we are going to assume the country codes recorded in variables `from` and `to` are 1, 2, …, 196 and that the same codes are recorded in variable `_ID` of `exports.dta`. If the true country codes were not 1, 2, …, 196, you could easily construct such country codes.

We are going to create a spatial weighting matrix from these data. Potential spillover from j to i will be

$$W_{i,j} = \frac{(\text{exports from } i \text{ to } j) + (\text{exports from } j \text{ to } i)}{i\text{'s GDP}}$$

This weighting matrix is a near cousin to one developed by Badinger and Egger (2008).

W would be easy to calculate if we had a matrix **E** recording exports and a vector **g** recording GDP. $E_{i,j}$ would be the exports from i to j, and g_i would record GDP of country i. The formula for $W_{i,j}$ would then be

$$W_{i,j} = (E_{i,j} + E_{j,i})/g_i$$

E and **g** can be easily created in Mata:

```
. tomata from to export gdp
. mata:
------------------------------------------------- mata (type end to exit) -----
: g = J(196, 1, 0)
: E = J(196, 196, 0)
: for (k=1; k<=length(exports); k++) {
>         i = from[k]
>         j = to[k]
>         g[i] = gdp[k]
>         E[i,j] = exports[k]
> }
: end
-------------------------------------------------------------------------------
```

`tomata` (Gould 2006) is a community-contributed command that makes it easy to create Mata matrix views of individual Stata variables. Type `search tomata` for details. The matrices will have the same names as the variables.

We can now calculate the weighting matrix:

```
. mata:
------------------------------------------------- mata (type end to exit) -----
: W = (E + E') :/ g
: end
-------------------------------------------------------------------------------
```

Finally, we post the result to Sp. We create column vector `id` containing 1, 2, …, 196 because row/column 1 of `W` corresponds to country code 1, row/column 2 of `W` corresponds to country code 2, and so on.

```
. mata:
------------------------------------------------- mata (type end to exit) -----
: id = 1::196
: end
-------------------------------------------------------------------------------

. spmatrix spfrommata Wt = W id
```

We could now use `exports.dta` and fit a model using `Wt` to create spatial lags.

References

Badinger, H., and P. H. Egger. 2008. Intra- and inter-industry productivity spillovers in OECD manufacturing: A spatial econometric perspective. CESifo Working Paper 2181, CESifo Group, Munich, Germany. http://www.cesifo-group.de/DocDL/cesifo1_wp2181.pdf.

Gould, W. W. 2006. Stata tip 35: Detecting whether data have changed. *Stata Journal* 6: 428–429.

Also see

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

[SP] **spmatrix create** — Create standard weighting matrices

[SP] **spmatrix matafromsp** — Copy weighting matrix to Mata

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

Mata Reference Manual

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark 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.