Intro 6 — Preparing data: Data without shapefiles or locations

Description Remarks and examples Also see

Description

This entry outlines the preparation of data without shapefiles or locations. Such data arise when spillover effects are based not on physical proximity but on proximity in other metrics.

You may also be interested in introductions to other aspects of Sp. Below, we provide links to those other introductions.

Intro 1	A brief introduction to SAR models
Intro 2	The **W** matrix
Intro 3	Preparing data for analysis
Intro 4	Preparing data: Data with shapefiles
Intro 5	Preparing data: Data containing locations (no shapefiles)
Intro 7	Example from start to finish
Intro 8	The Sp estimation commands

Remarks and examples

Remarks are presented under the following headings:

Nongeographic spatial data

Spatial analysis is about accounting for spillover effects. Consider an analysis of test scores of students. There may be spillover effects among friends for no other reason than friends share similar but relevant unmeasured characteristics. Or you might hypothesize more direct effects. Such data are known as social network data.

Consider the dollar value of trade between countries. Effects may spillover from one country to the next based on closeness measured by industry and the development level. Closeness might be based on the dissimilarity of industry (providing a reason to trade) and similarity of development level.

In these cases, the construction of the **W** spatial weighting matrices is often a substantive research problem in and of itself. As a result, researchers share weighting matrices. If you are analyzing such data, see [SP] **spmatrix import**. If you create such matrices, see [SP] **spmatrix userdefined**, [SP] **spmatrix fromdata**, [SP] **spmatrix spfrommata**, and [SP] **spmatrix export**.

First, however, you must prepare the data for use by Sp.

Preparation of cross-sectional data

We will assume that you have a dataset named `project_cs3.dta` that contains observations on nodes with variable `node_id` containing the standard codes for them.

To turn `project_cs3.dta` into Sp data, do the following:

Step 1: Load the data

```
. use project_cs3, clear
```

Step 2: Verify that `node_id` is an ID variable

```
. assert node_id!=.
. bysort node_id: assert _N==1
```

Step 3: `spset` the data

```
. spset node_id
```

Step 4: Save the data

```
. save, replace
```

In step 3, when we `spset` the data, `spset` created the new variable `_ID` containing a copy of the values in `node_id`. Variables `_CX` and `_CY` will not be created as they were in [SP] **Intro 4** and [SP] **Intro 5**, because these data do not contain location information.

In step 4, we save `project_cs3.dta` over itself. The new dataset differs from the old in that it has a new variable and it is `spset`. No changes or deletions were made to the data.

Preparation of panel data

We will now assume that you have `project_panel3.dta`, which is a panel dataset based on `node_id` and `time`.

To turn `project_panel3.dta` into Sp data, do the following:

Step 1: Load the dataset

```
. use project_panel3, clear
```

Step 2: Verify that `node_id` and `time` are jointly an ID variable

```
. assert node_id!=.
. assert time!=.
. bysort node_id time: assert _N==1
```

Step 2a: `xtset` the data

```
. xtset, clear
. xtset node_id time
```

Step 3: Balance and `spset` the data

```
. spbalance
. spset node_id
```

Step 4: Save the data

```
. save, replace     or     save newfilename
```

Concerning step 4, type `save, replace` only if step 3 did not involve dropping data.

There are no rules as there are with shapefiles

There are no special rules for working with the data created here as there were when working with data and shapefiles. The rules in [SP] **Intro 4** arose because of the linkage between the data file and its `*_shp.dta` file.

Also see

[SP] **spbalance** — Make panel data strongly balanced

[SP] **spset** — Declare data to be Sp spatial data