

## intro 5 — Preparing data: Data containing locations (no shapefiles)

[Contents](#)   
 [Description](#)   
 [Remarks and examples](#)   
 [Also see](#)

## Contents

---

<a href="#">intro 1</a>	A brief introduction to SAR models
<a href="#">intro 2</a>	The $W$ matrix
<a href="#">intro 3</a>	Preparing data for analysis
<a href="#">intro 4</a>	Preparing data: Data with shapefiles
<b><a href="#">intro 5</a></b>	<b>Preparing data: Data containing locations (no shapefiles)</b>
<a href="#">intro 6</a>	Preparing data: Data without shapefiles or locations
<a href="#">intro 7</a>	Example from start to finish
<a href="#">intro 8</a>	The Sp estimation commands

---

## Description

If you have data that already contain the coordinates of the geographical units, you can skip the shapefiles discussed in [SP] [intro 4](#). You are not required to skip them, however. Without shapefiles, you cannot create contiguity weighting matrices (matrices in which spillovers occur only among adjacent places), nor can you draw choropleth maps.

## Remarks and examples

stata.com

Remarks are presented under the following headings:

[Preparation of cross-sectional data](#)  
[Preparation of panel data](#)  
[There are no rules as there are with shapefiles](#)

## Preparation of cross-sectional data

We will assume that you have file `project_cs2.dta`, which is a cross-sectional dataset on U.S. counties over time, variable `fips` containing the standard county codes, and variables `locx` and `locy` identifying the location of each county.

To turn `project_cs2.dta` into Sp data, do the following:

Step 0: Load the dataset

```
. use project_cs2, clear
```

Step 1: Verify that `fips` is an ID variable

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

Step 2: `spset` the data

```
. spset fips, coord(locx locy)
```

Step 3: Set the coordinate units, if necessary

```
. spset, coordsys(latlong, miles)
```

Step 4: Save the data

```
. save, replace
```

That is all there is to it.

In step 2, we specified option `coord(locx locy)`. `spset` will create new variables `_ID`, `_CX`, and `_CY`. It will copy `fips` into `_ID`, and `locx` and `locy` into `_CX` and `_CY`.

In step 3, we set the coordinate system to degrees latitude and longitude because that was necessary in this case. We discussed in [SP] [intro 4](#) how to determine the coordinate system.

In step 4, we saved `project_cs2.dta` over itself. The new dataset differs from the old in that it has three new variables and is `spset`. No changes or deletions were made to the data.

## Preparation of panel data

This time, suppose `project_panel2.dta` is a panel dataset on U.S. counties over time. Perhaps it is already `xtset` on `fips` and `time`. The dataset also includes variables `locx` and `locy` identifying the location of each county.

To turn `project_panel2.dta` into Sp data, do the following:

Step 0: Load the dataset

```
. use project_panel2, clear
```

Step 1: Verify that `fips` and `time` jointly identify the observations

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

Step 1a: `xtset` the data and verify that `locx` and `locy` are constant within panel

```
. xtset, clear
. xtset fips time
. bysort fips (time): assert locx == locx[1]
. bysort fips (time): assert locy == locy[1]
```

Step 2: Balance and `spset` the data

```
. spbalance
. spset fips, coord(locx locy)
```

Step 3: Set the coordinate units, if necessary

```
. spset, coordsys(latlong) // optional
```

Step 4: Save the data

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

Concerning step 4, type `save`, `replace` only if step 2 did not involve dropping data.

In step 2, we `spset` the data, but not before verifying that they are strongly balanced. If the data are not strongly balanced, `spbalance` will issue an error and suggest that you type

```
. spbalance, balance
```

If you type that, `spbalance` will balance the data.

Then we `spset` the data. This creates the new variables `_ID`, `_CX`, and `_CY`. `spset` copies `fips` into `_ID` and copies `locx` and `locy` into `_CX` and `_CY`.

In step 3, we set coordinate units to degrees latitude and longitude. We discussed how to determine coordinate units in [\[SP\] intro 4](#).

## 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 linked `*_shp.dta` file.

## Also see

[\[SP\] `spbalance`](#) — Make panel data strongly balanced

[\[SP\] `spset`](#) — Declare data to be Sp spatial data