

EBERHARD KARLS  
UNIVERSITÄT  
TÜBINGEN



How to do a thematic map with Stata  
(Using Shapefiles)

Chair of Economic History

# Manual: Step by step

## 1. Install required commands

- To be able to use a (downloaded) shapefile in Stata and adjust the map according to certain data the following commands must be installed using the corresponding commands:
  - `ssc install spmap`
  - `ssc install shp2dta`

## 2. Find a shapefile online and download it

- The map that you want to fill or adjust later with your data must be downloaded in the first place since Stata does not automatically come along with such maps. There are lots of sources on the internet which provided these so-called “shapefiles” containing maps for (single) countries, as well as regions or the whole world.
- These maps can be found via search engines. Type in (for example) “shapefile whole world”. One webpage providing shapefiles (being used in the following example) is <https://www.naturalearthdata.com/downloads/>
- Even though the download contains (usually) lots of file, only two of them are needed, namely those ending on:
  - `.shp`
  - `.dbf`
- These two must be stored (together with the do-file) in the working directory

## 3. Translate the required files (using shp2dta)

- In a next step, the downloaded files must be translated in order to work with it.
- This is achieved with the command `shp2dta` which must be used in the following way:
  - `shp2dta using NAMEOFFILE, database(world_database) coordinates(world_coord) genid(id)`
- *NAMEOFFILE*: indicates the name of the files from section 2 (ending with `.shp` and `.dbf`).
- *database(world\_database)*: specifies that the file containing the database has the name “world\_database”. The name of the database can be used freely, but needs to be consistently used in the following steps.
- *coordinates(world\_coord)*: specifies that the file containing the database has the name “world\_coord”. Again, the name of the database can be used freely, but needs to be consistently used in the following steps.

- *gen(id)*: generates a numeric identifier which is called “id” and is unique for each country included in the map. Again, the name of the identifier can be use freely, but needs to be consistently used in the following steps.
- As a result, two new files “world\_database” and “world\_coord” are created into the working directory

#### 4. Examine the variables and match the identifier

- This step is necessary to bring the maps and your data which should be displayed in the map together.
- To examine in a first step which variables can be used to identify the countries of the map the following steps are necessary:
  - Tell Stata that you want to use the created database file (here: “world\_database”) with the following command: use world\_database, clear
  - Get an overview of the used variables with the following command: describe
  - Normally, the maps include a variable which is called NAME (or name/Name) which allows you to identify certain countries
  - To examine which country can be identified by the above created identifier (here: “id”), list all countries with the corresponding id, using the following command (with the variable name indicating the countries’ names): list NAME
- Now that the identifier for the countries are known, make sure that your data set also include a column called “id” (the name you chose above for the identifier), that it is saved in the Stata Dataset format. This step is important since without a common identifier the following steps cannot be performed.

#### 5. Merge the data sets

- Tell Stata that you know want to use the actual datafile that should be displayed in the map using the command: use NAMEOFFILE
- Now merge the two data sets using the following command: merge 1:1 id using world\_database
  - Here the identifier (“id”) is used to match the data of the datafile containing the data which should be displayed in the map, with the above created database “world\_database”

#### 6. Create the (final) map

- The map can be created using the following statement
  - `spsmap VARIABLE using world_coord, id(id) fcolor(Blues)`
- *VARIABLE*: indicates the variable of the data set which you want to display in the map

- *world\_coord*: is the above created file containing the coordinates of the map
- *fcolor(Blues)*: indicates which colors are used in the map to display the characteristic of the VARIABLE

### Practical Example:

- Problem:

An excel-file (see below) contains data on a certain variable (“Variable”) which is needed to be displayed in a map

|    | A                        | B  | C        |
|----|--------------------------|----|----------|
| 1  | Country                  | id | Variable |
| 2  | Fiji                     |    | 1        |
| 3  | Tanzania                 |    | 1,5      |
| 4  | W. Sahara                |    | 2        |
| 5  | Canada                   |    | 2,5      |
| 6  | United States of America |    | 3        |
| 7  | Kazakhstan               |    | 3,5      |
| 8  | Uzbekistan               |    | 4        |
| 9  | Papua New Guinea         |    | 4,5      |
| 10 | Indonesia                |    | 5        |
| 11 | Argentina                |    | 5,5      |
| 12 | Chile                    |    | 6        |
| 13 | Dem. Rep. Congo          |    | 6,5      |
| 14 | Somalia                  |    | 7        |
| 15 | Kenya                    |    | 7,5      |
| 16 | Sudan                    |    | 8        |
| 17 | Chad                     |    | 8,5      |
| 18 | Haiti                    |    | 9        |
| 19 | Dominican Rep.           |    | 9,5      |
| 20 | Russia                   |    | 10       |
| 21 | Bahamas                  |    | 10,5     |

- Result:
- Necessary steps:

1) Install the required commands:

```
Command
ssc install spmap
ssc install shp2dta
```

2) Download a shapefile and copy-paste the two files ending with .shp and .dbf into the working directory

Natural Earth

Free vector and raster map data at 1:10m, 1:50m, and 1:110m scales

Home Features Downloads Blog Forums Corrections About

## Downloads


Data themes are available in three levels of detail. For each scale, themes are listed on Cultural, Physical, and Raster category pages.

Stay up to date! Know when a new version of Natural Earth is released by subscribing to our [announcement list](#).

Overwhelmed? The [Natural Earth quick start kit](#) (227 mb) provides a small sample of Natural Earth themes styled in an ArcMap .MXD document and in a QGIS document. Download all vector themes as [SHP](#) (279 mb), [SQLite](#) (222 mb), or [GeoPackage](#) (260 mb).

Natural Earth is the creation of many [volunteers](#) and is supported by [NACIS](#). It is free for use in any type of project. [Full Terms of Use >](#)

**Large scale data, 1:10m**




[Cultural](#) [Physical](#) [Raster](#)

The most detailed. Suitable for making zoomed-in maps of countries and regions. Show the world on a large wall poster.

1:10,000,000  
1" = 458 miles  
1 cm = 100 km

**Medium scale data, 1:50m**




[Cultural](#) [Physical](#) [Raster](#)

Suitable for making zoomed-out maps of countries and regions. Show the world on a tabloid size page.

1:50,000,000  
1" = 790 miles  
1 cm = 500 km

**Small scale data, 1:110m**



[Cultural](#) [Physical](#)

Suitable for schematic maps of the world on a postcard or as a small locator globe.

1:110,000,000  
1" = 4,736 miles  
1 cm = 1,100 km

Tutorial\_Stata > Example\_MAP > Shapefile\_WholeWorld

Shapefile\_WholeWorld durchsuchen

| Name                              | Änderungsdatum   | Typ           | Größe  |
|-----------------------------------|------------------|---------------|--------|
| ne_110m_admin_0_countries.cpg     | 18.02.2019 13:42 | CPG-Datei     | 1 KB   |
| ne_110m_admin_0_countries.dbf     | 18.02.2019 13:42 | DBF-Datei     | 367 KB |
| ne_110m_admin_0_countries.prj     | 18.02.2019 13:42 | PRJ-Datei     | 1 KB   |
| ne_110m_admin_0_countries.README  | 18.02.2019 13:42 | HTML-Dokument | 24 KB  |
| ne_110m_admin_0_countries.shp     | 18.02.2019 13:42 | SHP-Datei     | 177 KB |
| ne_110m_admin_0_countries.shx     | 18.02.2019 13:42 | SHX-Datei     | 2 KB   |
| ne_110m_admin_0_countries.VERSION | 18.02.2019 13:42 | Textdokument  | 1 KB   |

| Name                          | Änderungsdatum   | Typ                   | Größe  |
|-------------------------------|------------------|-----------------------|--------|
| Shapefile_WholeWorld          | 19.02.2019 16:48 | Dateiordner           |        |
| Input_Map                     | 19.02.2019 16:40 | Microsoft Excel-Ar... | 9 KB   |
| ne_110m_admin_0_countries.dbf | 18.02.2019 13:42 | DBF-Datei             | 367 KB |
| ne_110m_admin_0_countries.shp | 18.02.2019 13:42 | SHP-Datei             | 177 KB |

3) Translate the required files

```
Command
shp2dta using ne_110m_admin_0_countries, database(world_database) genid(id) coordinates(world_coord)
```

4) Examine the required files and match the identifier

```

> orld_coord)
type: 5

. use world_database, clear

.
. describe

Contains data from world_database.dta
obs:      177
vars:     95          19 Feb 2019 16:53
size:    353,115

```

| variable name | storage type | display format | value label | variable label |
|---------------|--------------|----------------|-------------|----------------|
| featurecla    | str15        | %15s           |             | featurecla     |
| scalerank     | byte         | %10.0g         |             | scalerank      |
| LABELRANK     | byte         | %10.0g         |             | LABELRANK      |
| SOVEREIGNT    | str32        | %32s           |             | SOVEREIGNT     |
| SOV_A3        | str3         | %9s            |             | SOV_A3         |
| ADMO_DIF      | byte         | %10.0g         |             | ADMO_DIF       |
| LEVEL         | byte         | %10.0g         |             | LEVEL          |
| TYPE          | str17        | %17s           |             | TYPE           |
| ADMIN         | str35        | %35s           |             | ADMIN          |
| ADMO_A3       | str3         | %9s            |             | ADMO_A3        |
| GEOU_DIF      | byte         | %10.0g         |             | GEOU_DIF       |
| GEOUNIT       | str35        | %35s           |             | GEOUNIT        |
| GU_A3         | str3         | %9s            |             | GU_A3          |
| SU_DIF        | byte         | %10.0g         |             | SU_DIF         |
| SUBUNIT       | str35        | %35s           |             | SUBUNIT        |
| SU_A3         | str3         | %9s            |             | SU_A3          |
| BRK_DIFF      | byte         | %10.0g         |             | BRK_DIFF       |
| NAME          | str24        | %24s           |             | NAME           |
| NAME_LONG     | str35        | %35s           |             | NAME_LONG      |
| BRK_A3        | str3         | %9s            |             | BRK_A3         |
| BRK_NAME      | str32        | %32s           |             | BRK_NAME       |

```
. list id NAME
```

|     | id | NAME                     |
|-----|----|--------------------------|
| 1.  | 1  | Fiji                     |
| 2.  | 2  | Tanzania                 |
| 3.  | 3  | W. Sahara                |
| 4.  | 4  | Canada                   |
| 5.  | 5  | United States of America |
| 6.  | 6  | Kazakhstan               |
| 7.  | 7  | Uzbekistan               |
| 8.  | 8  | Papua New Guinea         |
| 9.  | 9  | Indonesia                |
| 10. | 10 | Argentina                |
| 11. | 11 | Chile                    |
| 12. | 12 | Dem. Rep. Congo          |
| 13. | 13 | Somalia                  |
| 14. | 14 | Kenya                    |
| 15. | 15 | Sudan                    |
| 16. | 16 | Chad                     |
| 17. | 17 | Haiti                    |
| 18. | 18 | Dominican Rep.           |
| 19. | 19 | Russia                   |
| 20. | 20 | Bahamas                  |

(Now that you know which identifier (“id”) belongs to which country, you can add it to your dataset, and it needs to be saved including the id before the next step can be performed.)

|    | A                        | B  | C        |
|----|--------------------------|----|----------|
| 1  | Country                  | id | Variable |
| 2  | Fiji                     | 1  | 1        |
| 3  | Tanzania                 | 2  | 1,5      |
| 4  | W. Sahara                | 3  | 2        |
| 5  | Canada                   | 4  | 2,5      |
| 6  | United States of America | 5  | 3        |
| 7  | Kazakhstan               | 6  | 3,5      |
| 8  | Uzbekistan               | 7  | 4        |
| 9  | Papua New Guinea         | 8  | 4,5      |
| 10 | Indonesia                | 9  | 5        |
| 11 | Argentina                | 10 | 5,5      |
| 12 | Chile                    | 11 | 6        |
| 13 | Dem. Rep. Congo          | 12 | 6,5      |
| 14 | Somalia                  | 13 | 7        |
| 15 | Kenya                    | 14 | 7,5      |
| 16 | Sudan                    | 15 | 8        |
| 17 | Chad                     | 16 | 8,5      |
| 18 | Haiti                    | 17 | 9        |
| 19 | Dominican Rep.           | 18 | 9,5      |
| 20 | Russia                   | 19 | 10       |
| 21 | Bahamas                  | 20 | 10,5     |

5) Merge the data sets

```
Command
use Input_Map

merge 1:1 id using world_database
```

6) Create the map

```
Command
spmap Variable using world_coord, id(id) fcolor(Blues)
```