

Published on **Development Impact**

Making visually appealing maps in Stata: Guest post by Asjad Naqvi

DEVELOPMENT IMPACT GUEST BLOGGER | FEBRUARY 07, 2022

This page in: English

Asjad Naqvi is an Assistant Professor at the Department of Socioeconomics, Vienna University of Economics and Business (WU), and frequently writes about data visualization on his [The Stata Guide on Medium](#) blog.

The ability to make maps in Stata is not new, but often, when one sees Stata maps online, little effort has been put into making them visually pleasing. The newer versions of Stata, together with more recent packages now allow for comprehensive map customizations. This post shows how to make a simple map in Stata, and then discusses how to improve its overall design.

Constructing a basic map in Stata

The core package for drawing maps in Stata is [spmap](#), written by Maurizio Pisati. To install this program, type:

```
ssc install spmap, replace
```

In order to use this package, you then need to have all the files that define the map layout and its attributes. These are known as *vector layers*, and can include points (e.g. location of a city or landmark), lines (e.g. roads) and polygons (e.g. shapes of regions like states or countries). The most common format for this information to be stored in is a *shapefile*, defined by [ESRI](#) (Environmental Systems Research Institute), the makers of ArcGIS, a widely used commercial mapping software. Each shapefile is a collection of several files, of which the following two contain the core set of information:

shp: contains the coordinates for the vector layer (e.g. boundaries of countries)

.dbf: contains the attributes of each map element (e.g. population or GDP of each country)

In addition to these two, there are usually several auxiliary files as well, the most important of which is *.prj*, or the the projection file, which stores information on the coordinate system of the vectors (see the FAQ below for more information on projections).

As an example, we can download the official World Bank country boundaries shapefile [here](#). Create a folder, and extract the file into this folder, and you will see a bunch of files all with the prefix *WB_countries_Admin0_10m*.



You then need to translate the *.shp* and *.dbf* files into Stata format. We can do this using the built-in *spshape2dta* command (introduced in version 15). For example, using the World Bank country boundaries dataset, change to the directory where you have unzipped the files and type:

```
spshape2dta WB_countries_Admin0_10m, replace saving(world)
```

This creates two Stata datasets: *world_shp.dta* which corresponds to the *.shp* file that contains the vector outlines, and *world.dta* that corresponds to the *.dbf* file and contains country-level information, for example, World Bank country classifications, income groups, GDP, population levels, etc. You can open each of these in Stata and explore their contents in the data browser.

Since the attributes data file, *world.dta* is at the country-level, you can add other country-level variables to this file as well. You just need to make sure that any data from other files is a 1:1 merge so that the country-level information remains unique. The *spshape2dta* command creates its own unique identifier variable *_ID* that is also used by the *spmap* command. Do not modify or change this variable!

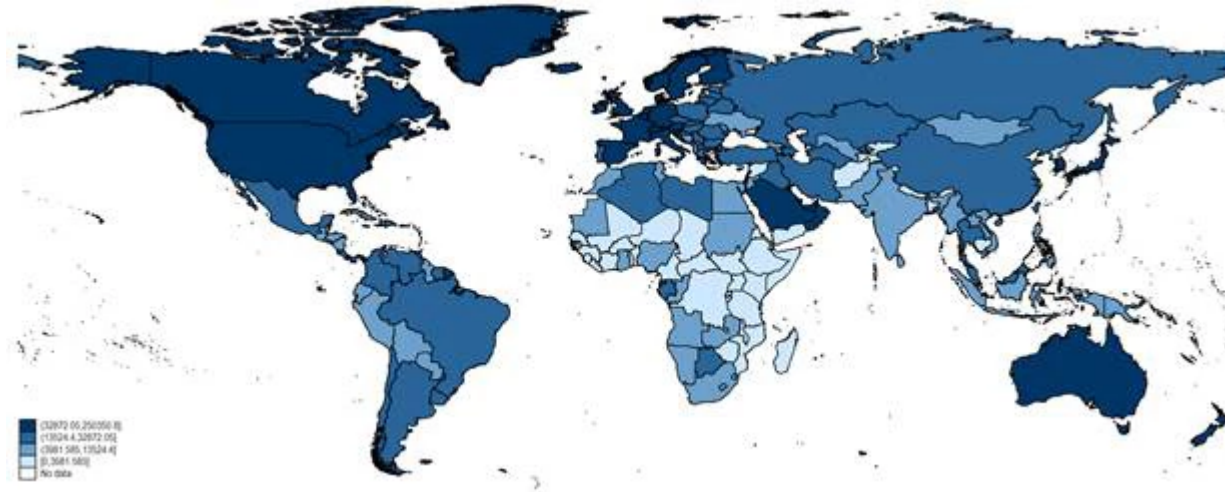
Since the World Bank shapefile already has a few variables included as part of the dataset, we will just use them to create GDP per capita as follows:

```
gen gdp_pc = (GDP_MD_EST / POP_EST) * 1000000
```

Then, to plot a basic map, we can use the *spmap* command as follows:

```
spmap gdp_pc using world_shp, id(_ID) fcolor(Blues)
```

This produces this default map below:



The spmap command has a lot of options to customize this map including several built-in color schemes. You can explore these but checking the help file (`help spmap`)

Making the map more visually appealing

Looking at the map above, we see several features of the map that we might want to change – for example, we might want to change the default projection to the more common Mercator projection that makes countries in Western Europe easier to see, and we might want to show more categories in the legend to highlight the variations. Additionally the legend is very small and for our variable that we are plotting, we don't really need all those decimal points.

Lastly, we might want different colors to help us differentiate the regions more. In order to do all of this, I recommend installing two additional packages: [geo2xy](#) (for changing projections) and [colorpalette](#) (for customizing colors):

```
ssc install geo2xy, replace  
  
ssc install palettes, replace  
  
ssc install colrspace, replace
```



As an optional step, you can also install the [schemepack](#) package if you want a clean look for your graphs:

```
ssc install schemepack, replace
```

And change to a minimalistic scheme:

```
set scheme white_tableau
```

The pack contains a lot of different schemes that you can explore in its help file (help schemepack).

Next, we change the projection to a Mercator projection and save it under a new file name:

```
use world_shp, clear  
  
// Make sure the coordinates are inside the (-180,180)  
bounds
```

```
replace _X = 180 if _X > 180 & _X!=.  
geo2xy _Y _X, proj(web_mercator) replace  
save world_shp2.dta, replace
```

Then we might want to use the `colorpalettes` package to change the color styles we use and how many intervals we define, and other options of the `spmap` to control where the intervals are defined, where we plot the legend and put a little box around it, change the thickness of the country boundaries and perhaps put them in white (using `spmap`'s `osize` and `ocolor` options), and give the map a title and notes:

```
* load the data again if you have not saved it previously
```

```
use world, clear
```

```
gen gdp_pc = (GDP_MD_EST / POP_EST) * 1000000
```

```
format gdp_pc %12.0fc
```

```
* generate the colors
```

```
colorpalette viridis, n(12) nograph reverse
```

```
local colors `r(p)'
```

```
* generate the map and pass the colors and other modifications
```

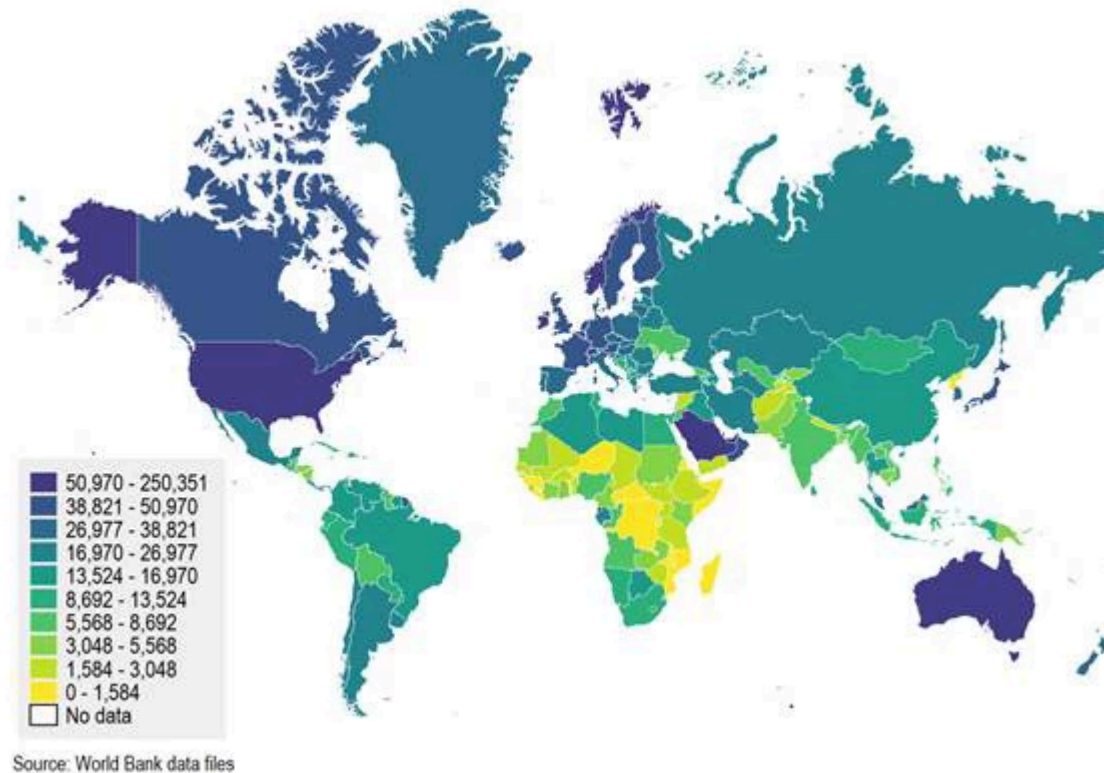
```
spmap gdp_pc using world_shp2, id(_ID) ///  
  
    clnum(10) ///  
  
    fcolor("`colors'")    ///  
  
                                legstyle(2)    legend(pos(7)    size(2.8)  
region(fcolor(gs15)))    ///  
  
    ocolor(white ..) osize(0.05 ..)    ///  
  
    title("My first customized map in Stata: GDP per capita",  
size(5))    ///  
  
    note("Source: World Bank data files", size(2.5))
```

The important parts of the code are highlighted in bold. The format option %12.0fc means to show up to 12 digits and no decimals, while the “c” tells Stata to show the 1000 separator in the number is commas.

We generate the colors using the colorpalette package. Here we generate 12 colors instead of 10. We do this to stretch the darker colors out of the 10 categories. Feel free to play around with these options. The colors are stored in a local which is later passed on to the spmap package. Here we draw the map using the world_shp2 file, define 10 color cut-offs, use legend style (legstyle) option 2, which adds a separator – for each legend item. We also increase its size and give it a grey background color.

From the code above, we get the following map:

My first customized map in Stata: GDP per capita

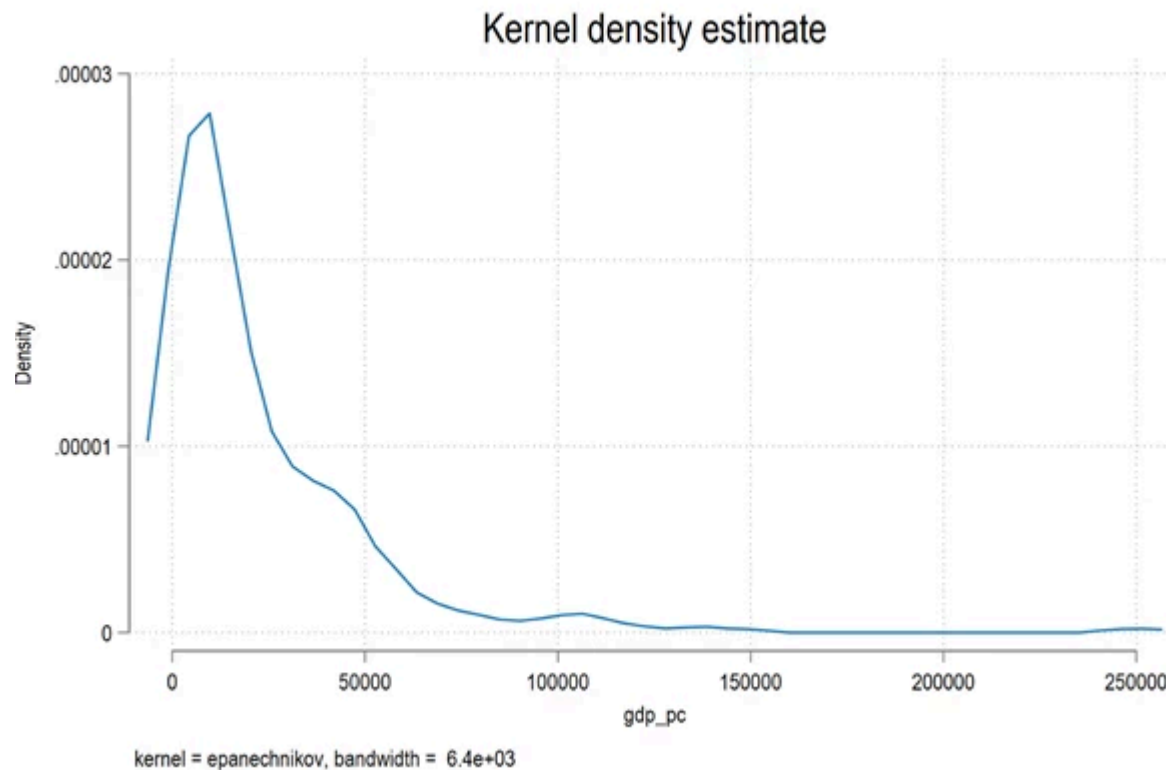


Here we can see that just a few extra lines of code changes the feel of the map completely. A [dofile on my GitHub](#) provides more examples to illustrate the use of these map options.

Some tips when making maps

Maps are about story telling. Therefore, when making them, make sure that the key takeaway is immediately visible. Here are some tips to keep in mind to help you achieve this:

Cut-offs: Explore the data you are plotting. For example, in your example above, we plot GDP per capita. We can look at its distribution by typing `kdensity gdp_pc`:



Note that the distribution is not evenly split, nor is it normally distributed. Thus if we use equal intervals, or deviations from the mean, then the map will not look exciting at all. You can try these out by changing the method (in the `spmap` helpfile, see the entry for `clmethod`). Instead we used the default “quantiles” option which splits the data based on percentiles. Even though, the quantiles option works well in most cases, it is still advisable to see how the map changes

when using other cut-off methods. Playing around with these is not unusual. Most of the maps we see online do have some subjective decision on how the categories are defined. Unless there is some strong reason to use one specific method, go with the one which highlights the variations the best.

There are also cases when auto-generated cut-offs don't work or should not be used. For example, if you are comparing the same variables across different regions or across time. Here, a full control over the cut-offs is essential. We want to make sure that values are assigned to the same color bins in order to see spatial-temporal changes.

In certain cases manual cut-offs are relatively easy to figure out. For example, if a variable is in percentage terms, or has well-defined bounds (like the Human Development Index which is in the 0-100 range), then we just bin the data in ranges of 10. In `spmap` this is easily done by using the "custom method" option `clm(custom) clbreaks(0(10)100)` which gives us easy to read 0-10, 10-20, 20-30,... bins.

Colors: Colors are an essential part of making maps. This is why the `colorpalette` package is so important. Not only does it provides a large selection of some of the best color templates out there, it also allows us to add our [own color schemes](#), change the color contrasts, intensity, and saturation, etc.

When it comes to colors, it is important to know what kind of data you are plotting. It can be sequential (increasing or decreasing in magnitude), diverging (some values are positive and some are negative), or discrete (categories are



mutually exclusive and not ordered, like in qualitative data). Each of these categories has its own set of color templates.

One also needs to consider whether the color schemes are colorblind friendly and black and white friendly. The color scheme we used above, Viridis, fulfills both these criteria and also provides a decent color contrast to make categories highly distinguishable. It is also one of the most used color schemes by professionals.

Legends: The next point one needs to care about are legend entries which help us read the maps. If your map is going in a paper or you are posting it online, then make sure that the legends have a decent enough size so that can be read easily. Most of the Stata maps one sees online use the defaults which does not work in all circumstances. So adapt these to the map requirements. Additionally, think about is whether you need all those decimals points. If not, then round off the numbers to some reasonable value.

Notes: Last point are notes. Annotating maps, and figures in general, is good practice. Add the map data sources, projection type, and how you defined the cut-offs, etc. One rarely sees these in maps, but it is good practice and helps with replicability.

Some F.A.Q.s about maps:

What are map projections?



The earth is not flat, and neither is it a perfect sphere. It is, in fact, an ellipsoid. And drawing this shape on a 2D surface is not trivial. Map projections are used to display maps and each projection has its own advantage. Projections themselves come in three broad categories: azimuth, conical, and cylindrical. Different map projections are used to calculate correct for distances, directions, shapes and areas. In other words, one projection fixes one of the above four issue and distorts the rest. There are of course many intermediate projections as well that try and preserve at least two properties, but they usually cover very small areas. Cylindrical projections are the most common types that we see. For example, the web Mercator projection, that we also used in the example above, is the most common one used in online maps but it also distorts the actual sizes for the sake of visibility.

Be careful when combining different files that you are using the same coordinate systems and projections.

The most common coordinate system is the [World Geodetic System](#) or WGS84, which is the global standard for taking GPS coordinates. Most shapefiles come in the WGS84 coordinate system and then projected. In general, when you use a spatial file, check its metadata first. This is really important especially if you want to modify the projection of a layer, or combine different files that are in different projections. This could include adding your own spatial data (e.g. GPS locations of survey villages).

If you are not sure how to deal with these, then a recommended way is to open the layers in softwares like ArcGIS or QGIS (free), which automatically detect



projections and align the layers together (most of the time!). These aligned layers can be exported again in a common coordinate system, which you can import in Stata in a second step.

What are the other types of formats?

In addition to ESRI shapefiles, vector data can also exist in JSON, geoJSON and other web-friendly formats which are typically used for online visualizations and interactive maps. Other formats include Google's Keyhole Markup Language (KML), Autocad DXF files, Vector Product Format (VPF) designed by the US Department of Defense, OSM designed by OpenStreetMap, Scalable Vector Graphics (SVGs), and many others. Most softwares are able to easily convert between these formats.

Working with administrative boundaries

Countries come in all shapes and sizes. Hence their subdivisions also vary significantly. Larger countries, especially those that are densely populated, can have several layers of administrative regions. Regional divisions can also have different names across the countries. For example, provinces, districts, urban and rural settlements, villages, etc. If you are working across many different countries of the world, there is a homogenized system referred to as Database of Global Administrative Areas or [GADM](#). This database, which is regularly updated, allows us to compare regional across countries regardless of what they are called. For example, GADM0 are country boundaries, GADM1 are the next divisions, and so on.

Where can I find spatial files?



It depends on your geographical focus. Most high income countries have their shapefiles available in several formats on their statistical agencies. For example, the [USA Census Bureau](#) provides it for the USA, and the [Eurostat](#) for European countries. For developing countries, I recommend going to the websites of international development organizations that collate this information for several countries. For example a good starting point is the [GADM database](#) where country-level datasets can be downloaded. Other resources include UN OCHA's [Humanitarian Data Exchange](#) (HDX) that contains administrative boundaries for a large set of countries. The [Berkeley Library's GeoData](#) service also has a large collection. Another open source is [DIVA-GIS](#) which provides several additional layers on top of the administrative boundaries including roads, river, elevation, etc. Some layers might be outdated though so check them carefully. Another place, where I have surprisingly found a lot of spatial data is [GitHub](#). Since maps and web development has picked up so much in the recent years, and GitHub is the main place for data scientists, it has a very large searchable database. If all the websites fail to return a satisfactory return, do check here as well! If you want detailed information on certain regions, the one of the best sources is [OpenStreetMaps](#) (OSM), that fully allows users to download data. You can also use OSM with third-party softwares, like QGIS to [extract these layers as shapefiles](#).

Accessing finer-level administrative boundaries or point locations within countries might be more challenging. First of all, boundaries are constantly evolving. So if your aim is accuracy, then check the sources carefully and try to get the information from the official statistical or the mapping agency of the



country. The finer the unit, the more difficult it is to find its most recent and accurate representation. In some cases, finer-scale data is also censored for various reasons including privacy and security, or just plain bureaucracy. As a note of caution, before starting spatial analysis at a finer level for a specific region, get as much information as possible, especially on what is legally allowed. However, if the aim of making your map is just depicting of choropleth or heatmaps, then accuracy of maps should not matter so much.

I want to learn more! Are there some nice resources online?

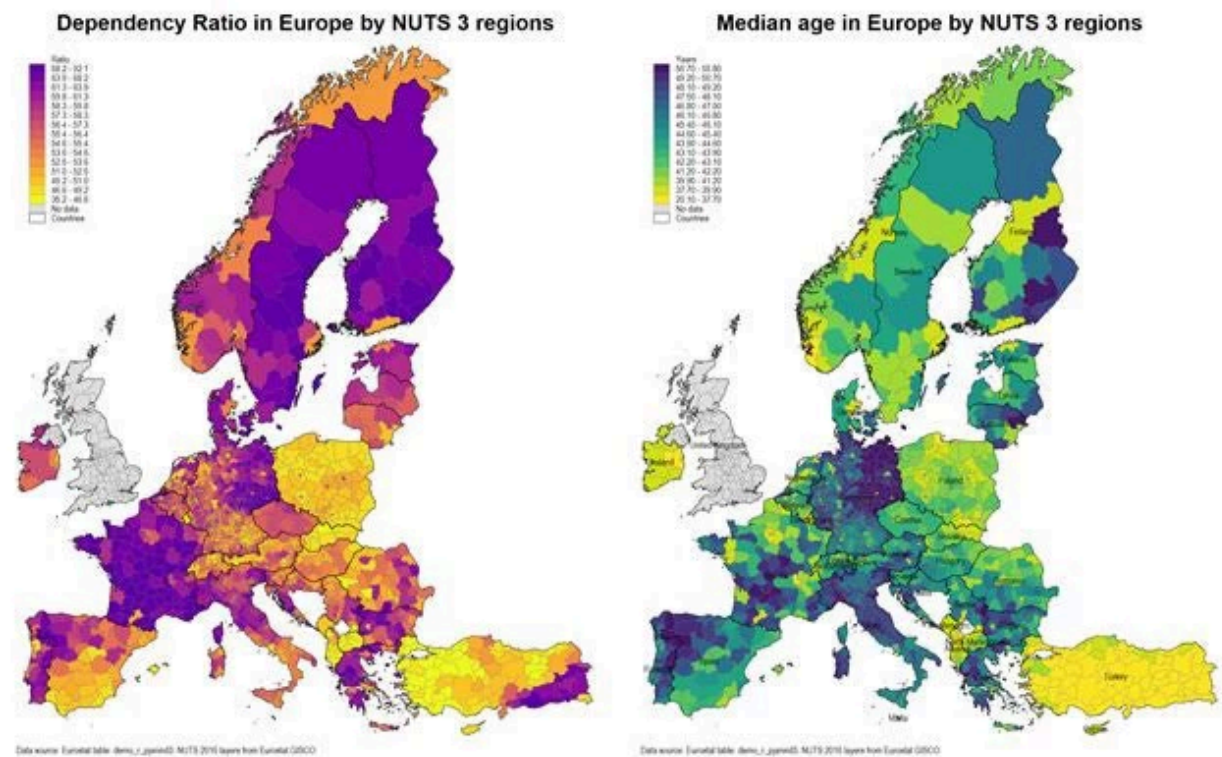
Yes! Luckily a lot more is being written on this topic now. Check out the official entries on the Stata blog [here](#). Also check out my [Stata Guide on Medium](#), especially the [Maps](#) section which has several entries that range from the basics of map making to more advanced application including fully customizing maps, integrating OpenStreetMaps data in Stata, and drawing maps from scratch.

Here are a few examples of the maps that you can learn to make by following the guides:

Heat maps with administrative boundaries:

The maps below show simple variation across European regions for two indicators: dependency ratio (retired and children as a ratio of working age population), and the median age. The maps highlight the use of color schemes for spatial clustering. They also show how different level of boundaries (regional and country-level) can be combined in one map including adding labels to countries. You can learn how to make these maps in the [Maps in Stata II](#) guide.

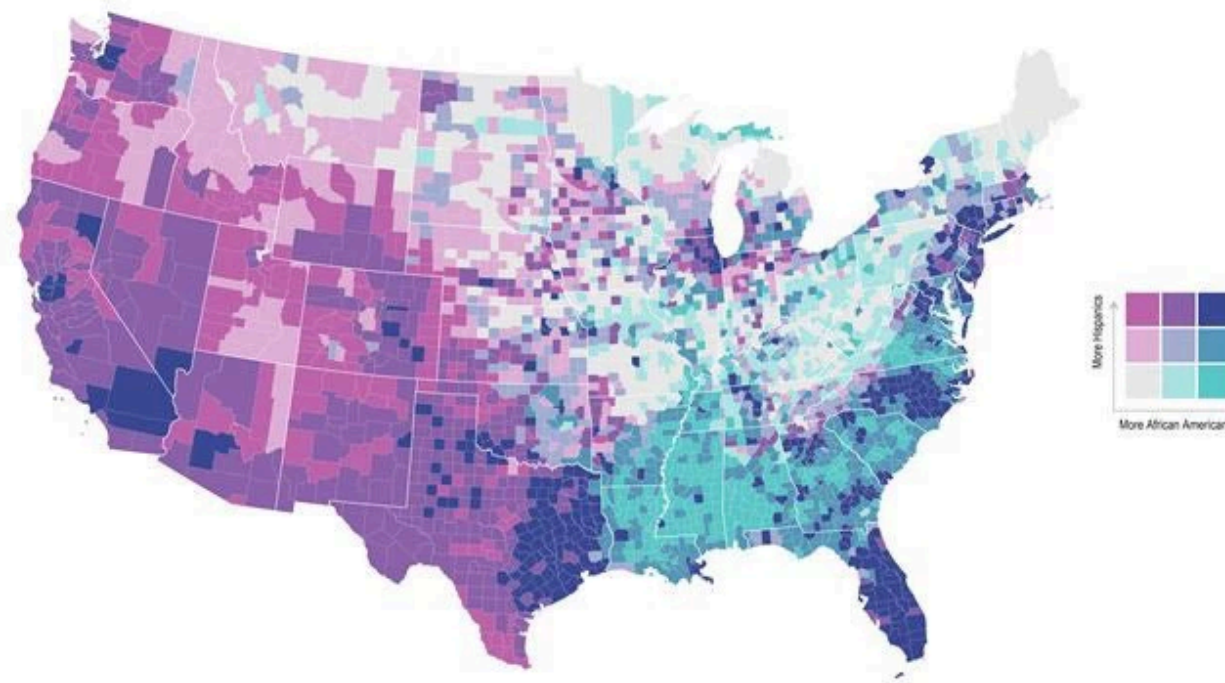




The bi-variate map below shows how two categories can be combined to highlight some interesting combinations. For example, darker shades of green show a higher share of African-Americans in county-level population of the USA, while darker shades of pink show more Hispanics. The map highlights spatial segregation of the two groups across the eastern and the western parts of the country. Some states like Florida, Texas, and New York, have a higher share of both. You can learn how to make this map in the [Stata Bi-variate maps guide](#).

Distribution of African-Americans and Hispanics in the USA

Colors defined by tercile cut-offs of race shares in counties



Data source: United States Census Bureau (<https://www.census.gov/>).

Even though, Stata cannot process spatial images or rasters, the information can still be extracted using softwares like ArcGIS or QGIS. The maps below show layers extracted from OpenStreetMaps using QGIS and exported to Stata for visualization. This allows complete flexibility for making any type of maps we want. The tutorial for making these maps is provided in the [Advanced Mapping with Stata](#) guide:





I hope these provide a taste of the map-making capabilities available in Stata. You can see other Stata maps I have made as part of the [2021 30DayMapChallenge](#). I also regularly write about Stata-related content, especially visualizations, on [The Stata Guide on Medium](#). Please feel to reach out if you have queries about code and making maps in Stata. I am also active on Twitter [@AsjadNaqvi](#).

Get updates from Development Impact

Enter your email to receive email notifications

SUBSCRIBE

Language: ☒ English

* ☐ I agree with the terms of the Privacy Notice and consent to my personal data being processed, to the extent necessary, to receive these updates.



Authors



Development Impact Guest Blogger

Guest bloggers

MORE BLOGS BY DEVELOPMENT

Join the Conversation

Your name

Your Email

The content of this field is kept private and will not be shown publicly

Write a response...

Remaining characters: **1000**

☐ I have read the [Privacy Notice](#) and consent to my personal data being processed, to the extent necessary, to submit my comment for moderation. I also consent to having my name published.

SAVE

