

(<https://www.thebrandonjackson.com/>)

SHAPEFILES IN MATLAB

📅 December 17, 2016 (<https://www.thebrandonjackson.com/shapefiles-in-matlab/>) 👤

Brandon (<https://www.thebrandonjackson.com/author/bjck1990/>)

A shapefile is a standard geospatial data format for storing location, shape, and attributes of geospatial features. The mapping toolbox in MATLAB has tools for working with these datasets. In this tutorial we will work with some of this functionality.

Shapefiles are typically composed of very large datasets. Shapefiles are actually composed of multiple required files that are shared in a compressed *.zip format. The three required files are:

- *.shp – The main file that stores the geometry
- *.shx – Stores index information for the geometry
- *.dbf – Stores attributes for each shape

More information about shapefiles can be found here (http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/What_is_a_shapefile/005600000002000000/).

Examples of shapefiles can be found at the following sources:

1. NOAA Storm Prediction Center Severe Weather GIS (SVRGIS) Page: <http://www.spc.noaa.gov/gis/svrgis/> (<http://www.spc.noaa.gov/gis/svrgis/>)
2. Natural Earth: <http://www.naturalearthdata.com/downloads/> (<http://www.naturalearthdata.com/downloads/>)
3. Free GIS Datasets: <https://freegisdata.rtwilson.com> (<https://freegisdata.rtwilson.com>)
4. US Energy Information Agency: https://www.eia.gov/maps/layer_info-m.php (https://www.eia.gov/maps/layer_info-m.php)

For this tutorial, I will use the tornado (torn.zip) and states (states.zip) datasets from NOAA SVRGIS (<http://www.spc.noaa.gov/gis/svrgis/>).

The following functions will be essential to this tutorial:

- shaperead (<https://www.mathworks.com/help/map/ref/shaperead.html>) – Reads in the shapefiles
- makesymbolspec (<https://www.mathworks.com/help/map/ref/makesymbolspec.html>) – Defines structure for plotting attributes
- mapshow (<https://www.mathworks.com/help/map/ref/mapshow.html>) – Enables the shape data to be displayed

First, we will start out by loading the shapefiles into MATLAB. We will use the shaperead function for this task. For large datasets, the shaperead function makes it easy to load only specific sections of the data based on from a certain value range. This functionality will not be necessary for this example.

```

1
2 % Load the shape files
3 Tornadoes = shaperead('torn.shp');
  States = shaperead('states.shp');
```

For this tutorial, we will color the tornado paths based on the magnitude of the tornado. There are two approaches to do this. The first uses the matlab makesymbolspec function. This function outputs a structure containing the symbolization specification:

```

05 |
06 | ColorMap = colormap(jet(6));
07 |
08 | TornSpec = makesymbolspec('Line',...
09 |   {'mag',0, 'Color',ColorMap(1,:)},...
10 |   {'mag',1, 'Color',ColorMap(2,:)},...
11 |   {'mag',2, 'Color',ColorMap(3,:)},...
   |   {'mag',3, 'Color',ColorMap(4,:)},...

```

For datasets containing a plethora of specifications, it may be desirable to assemble the data within a loop. This can be done for this example as follows:

```

05 |
06 | % Set colormap for Fujita scale
07 | ColorMap = colormap(jet(6));
08 | TornSpec.ShapeType = 'Line';
09 | for ii = 0:5
10 |   TornSpec.Color{ii+1,1} = 'mag';
11 |   TornSpec.Color{ii+1,2} = ii;
12 |   TornSpec.Color{ii+1,3} = ColorMap(ii+1,:);
   | end

```

Now we will plot the the tornado paths overlaid with the boundaries of the states.

```

14 |
15 | %% Plot the entire United States
16 | % Top Plot - entire country
17 | ax(1) = subplot(3,2,[1:4]);
18 | h_states = mapshow(States, 'FaceColor', [1 1 1]);
19 | h_torn = mapshow(ax(1), Tornadoes, 'SymbolSpec',TornSpec);
20 | axis(ax(1),'equal');
21 | ax(1).Xlim = [-2600000,2400000];
22 | ax(1).Ylim = [-1900000,1500000];
23 | title('Tornadoes (1950-2015)');
24 |
25 | % Legend Information, we will simulate a legend by create patch lines and
26 | % hiding them so they do not appear on the plot
27 | for ii = 0:5
28 |   v_patch = [0 0; 1 0; 1 1; 0 1];
29 |   f_patch = [1 2 3 4];
30 |   h_patch(ii+1) = patch('Faces',f_patch,'Vertices',v_patch,...
31 |     'FaceColor',ColorMap(ii+1,:), 'DisplayName', ['F' num2str(ii)]);
32 |   h_patch(ii+1).Visible = 'off';
33 | end
34 | h_legend = legend('toggle');
35 | h_legend.Location = 'south';
36 | h_legend.Orientation = 'horizontal';
   | axis off

```

We will also plot the number of tornadoes per year and the relative frequency of each tornado strength. I will not present the entire code to complete this, but I will give you enough to get started:

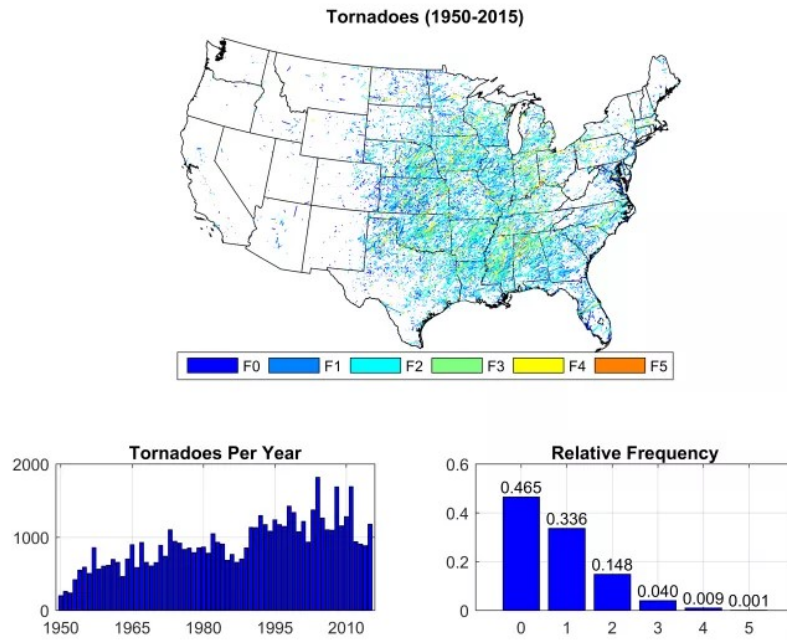
```

38 |
39 | %% Add secondary plots
40 | % Tornadoes Per Year
41 | ax(2) = subplot(3,2,5);
   | Years = [Tornadoes.yr];


```

The results of this plot can be viewed below.

(https://i1.wp.com/www.thebrandonjackson.com/wp-content/uploads/2016/12/Tornado_Map_Large.png)



You may notice a significant increase in the number of tornadoes per year starting around 1990. According to the NOAA storm prediction center, this increase is a result of an increased ability to detect tornadoes resulting from the adoption of Doppler Radar network (<http://www.spc.noaa.gov/efscale/>).

 MATLAB Tutorials (<https://www.thebrandonjackson.com/topics/matlab/>)

Comments are closed.