Generating Custom Shape Files for Data Visualization

Stephanie R. Thompson, Rochester Institute of Technology, Rochester, NY

ABSTRACT

There are times you may want to represent your data graphically but the existing options do not work. Using a custom shape file or even a file with coordinates you create yourself can get you what you want. This quick presentation will show you how.

INTRODUCTION

All it takes to create a custom visualization are two SAS datsets and one procedure. The first dataset is a type of shape file that contains the coordinates and identification of your shapes. The other contains the value to be represented visually. This value can be a population, amount, or any other number which will be used to set the scale. The procedure, PROC GMAP, will bring the 2 files together and add color.

THE SHAPES

The files in the SAS maps library are special shape files for creating a map. Many times these files are all you need to create your visualization. However, if you want to generate a map of a college campus or some other collection of shapes, you will either need to find this file or generate it yourself.

When existing map or shape files are lacking, it is very easy to create them yourself. The basic map file contains only 3 variables: x, y, and an id of some type, idnum in this example. The x and y define the coordinates of the shapes and the idnum delineates each shape. Without the idnum, the entire set of coordinates become connected like one continuous dot to dot.

A very basic shape file is created by the datastep below:

```
data shapes;
input x y idnum;
cards;
1     1     1
5     1     1
3     5     1
8     1     2
12    1     2
12    5     2
8     5     2
3     6     3
7     6     3
9     8     3
7     10    3
3     10    3
1     8     3
;
run;
```

Using cards (or datalines) is just one way of generating the file. Any SAS dataset with these three variables will work.

THE VALUES

The other dataset contains the idnum and a value to be used in the visualization. The idnum variable needs to match the one in the shapes file in terms of type and values. This example uses a numeric idnum, but it can also

be a character variable. As long as it is the same type in both files the visualization will render. The name of the variable is not important.

A values dataset could look like the following:

```
data values;
input idnum amt;
cards;
1   10
2   250
3   100
;
run;
```

PROC GMAP

The procedure which brings the shapes and values together is PROC GMAP. The syntax is very simple.

```
proc gmap data = values map = shapes;
id idnum;
choro amt / discrete;
run;
quit;
```

ID specifies the variable that delineates the individual shapes. CHORO in combination with the variable amt states which variable contains the values used to fill the map shapes. The DISCRETE option is used to generate one distinct color for each level of amt. Omitting the discrete option will generate a single color gradient distinguishing the levels of amt.

The QUIT statement is necessary after the run since PROC GMAP is one of the procedures that supports RUN-group processing. This means is that the procedure will execute but continue to run until either a QUIT statement or step boundary like a datastep is encountered.
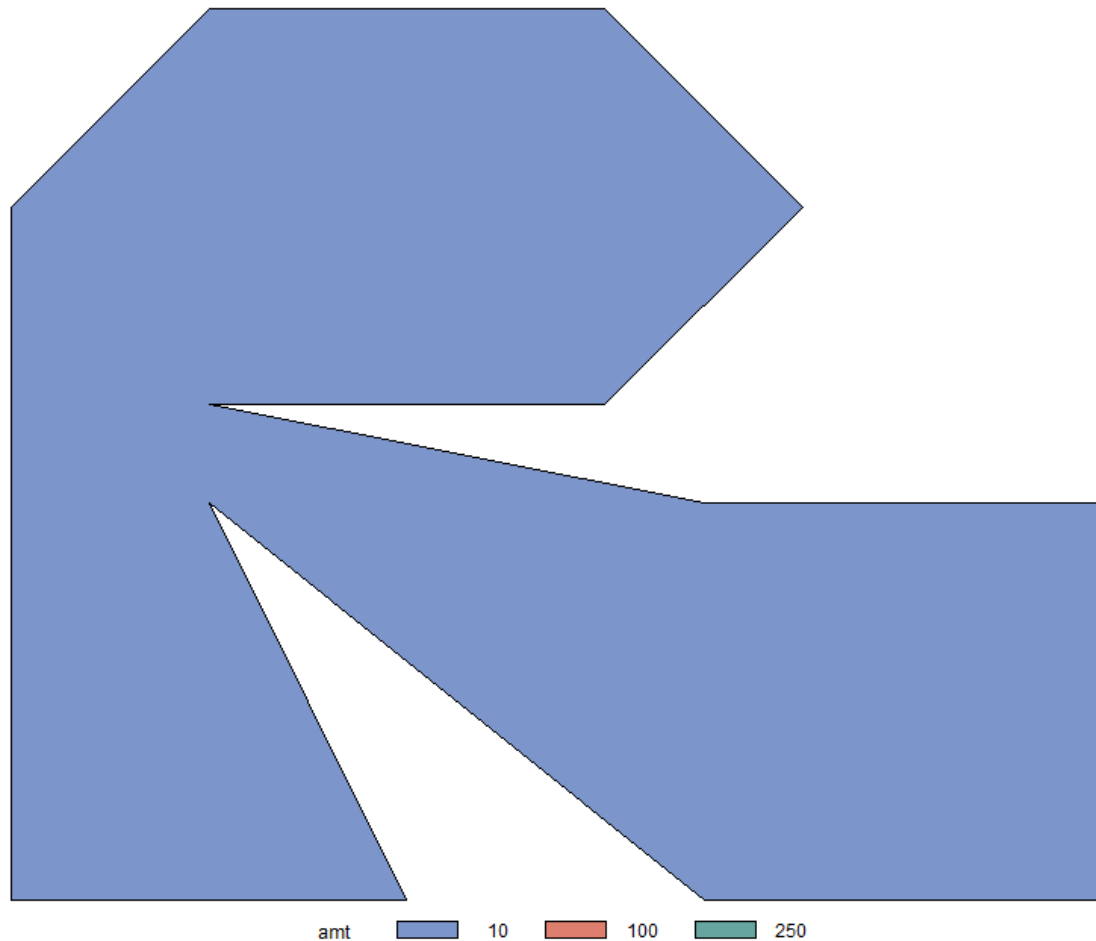
RESULTS

The visualization generated from the example code is below.
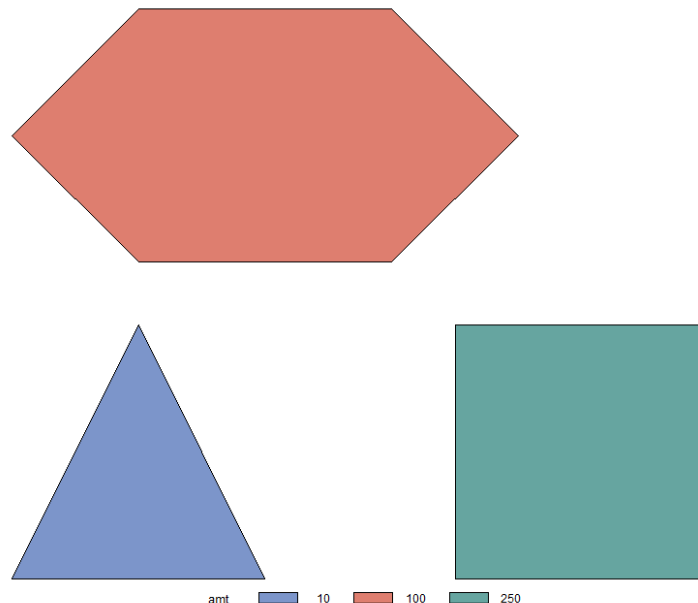
This is a very simple visualization but shows the basics for creating your own shapes.

A version without defining each shape is shown to give an idea of the dot to dot look of the shapes run together.

CONCLUSION

This very basic example shows how to create your own shape file to use for data visualization. The colors in the values dataset can be used for simple coloring or to show actual values.   This approach is useful for generating simple designs through complex maps.

ACKNOWLEDGMENTS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Stephanie R. Thompson
Rochester Institute of Technology
Email: Stephanie.Thompson@rit.edu
Web: http://finweb.rit.edu/irps/