

```
geo.graph.sizecoloroutline{slm}
```

## Size/Color Maps with Borders

### Description

Creates a map with color keyed to one variable, size keyed to another variable, lines showing neighbors, and the option of a map (polygon) outline.

### Usage

```
geo.graph.sizecoloroutline(data, rshape.outline = NULL,  
  xcol = xcol, ycol = ycol,  
  color.col, color.fun = eval, size.col, size.fun = eval,  
  cex.start.size = 1.5, cex.increment.size = .001,  
  dec.dig, graphic.device, output.filename)
```

### Arguments

<code>data</code>	a data frame containing the data.
<code>rshape.outline</code>	An object of class <code>map</code> that will be used to plot the polygons. Typically, this is created using <code>read.shapefile()</code> in the <code>maptools</code> package. The default is <code>NULL</code> , in which case only the spatial locations are plotted.
<code>xcol</code>	the column containing the x-coordinate values in the data frame. The default is a variable called <code>xcol</code> that contains the name of the <code>xcol</code> . If it is typed in directly, it should be quoted; e.g., <code>xcol = "x"</code> .
<code>ycol</code>	the column containing the y-coordinate values in the data frame. The default is a variable called <code>ycol</code> that contains the name of the <code>ycol</code> . If it is typed in directly, it should be quoted; e.g., <code>ycol = "y"</code> .
<code>color.col</code>	the name of the variable in <code>data</code> to be plotted using colors. It should be quoted, e.g., <code>"estimate"</code> . Ten color classes are created of equal intervals from the lowest to highest values. So far, the only color palette is, <code>"darkblue"</code> , <code>"blue"</code> , <code>"cyan3"</code> , <code>"cyan"</code> , <code>"lightgreen"</code> , <code>"greenyellow"</code> , <code>"yellow"</code> , <code>"orange"</code> , <code>"tomato"</code> , <code>"red"</code> .
<code>color.fun</code>	the name of a function to be applied to the variable with name <code>color.col</code> for the purposes of creating color classes. An example is <code>color.fun=log</code> .
<code>size.col</code>	the name of the variable in <code>data</code> to be plotted using different sizes. It should be quoted, e.g., <code>"std.err"</code> . Ten size classes are created of equal intervals from the lowest to highest values. Minimum and maximum symbol sizes are determined using <code>cex.start.size</code> and <code>cex.increment.size</code> .
<code>size.fun</code>	the name of a function to be applied to the variable with name <code>size.col</code> for the purposes of creating size classes. An example is <code>color.fun=log</code> .

`cex.start.size` the size of the first symbol class. The default is `cex = 1.5`.

`cex.increment.size` the increments between symbol class sizes. The default is 0.001, which will cause all symbols to be the same size.

`dec.dig` the number of decimal digits when printing the legend to the map.

`graphic.device` a graphic device will be started. The default is `graphic.device = "windows"`. The alternative is "postscript", which requires an output filename.

`output.filename` The output filename for the graphic device. It should be quoted; e.g., `output.filename = "d:\\mydata\\mygraph.ps"`

## Examples

```
# --- AN OMNIDIRECTIONAL EXAMPLE WITH 20 BINS OF SIZE 1000 METERS

geo.graph.sizecoloroutline(data = mydata, rshape.outline =
  import.rshape, xcol = "x", ycol = "y",
  color.col = "cv.resid", size.col = "cv.se", size.fun = invf,
  cex.start.size = 1, cex.increment.size = .2,
  dec.dig = 4, graphic.device = "postscript",
  output.filename = "d:\\mydata\\mygraph.ps")
```