Graph Set Up

Dave Lorenz

September 9, 2013

These examples demonstrate how to set up a complete figure beginning with a simple scatter plot. The general procedures apply to any other high-level graphics functions within the USGSwsGraphs package. All of the examples use randomly generated sets of data. **NOTE:** to use any high-level gaphics function in the USGSwsGraphs package, you must first call a function to set up the graphics environment like `setPage` or `setPDF`, but these are not included here to use the graphics tools in `Sweave`.

```
> # Load the USGSwsGraphs package
> library(USGSwsGraphs)
> # Generate the random data
> set.seed(27036)
> X <- rnorm(32)
> Y <- X + rnorm(32)
> Z <- rnorm(32, sd=1.2)
```

Some basic nomenclature is necessary to understand the instructions in this vignette.

figure The completed product, consisting of one or more graphs and an optional explanation.

graph The area containing the plot, axis labels and titles and graph title.

plot The data plotted as lines, points, or other.

axis labels Labels for each major tick.

axis title A brief desciption of the axis.

graph title A description or identifier for the graph.

explanation An explanation of the symbols in the graphs, also called legend or key.

1 A Simple Graph

The simplest graph can be created using only two functions, one to set up the graphics environment and one to create the graph. An additional call would be required to close the graph if the output were a PDF file created by `setPDF` or `setSweave`.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device would be setPage("square")
> setSweave("graph01", 6 ,6)
> #
> xyPlot(X, Y)
> # Required call to close PDF output graphics
> graphics.off()
```

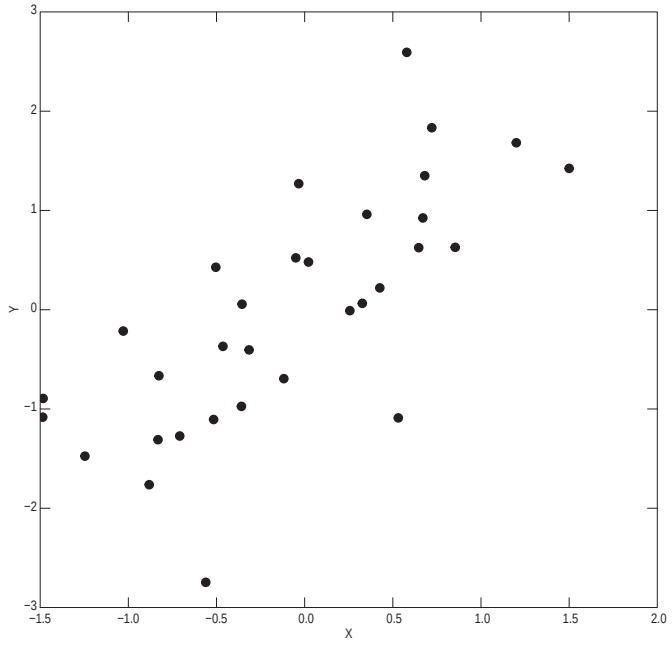

Figure 1. A simple x-y scatter plot.

2 Plot Customization

The default plot created by `xyPlot` is a simple x-y scatter plot. The axes are scaled to a range that fits the data and labeled according accepted practices within the USGS The x- and y-axis titles are derived from the names of the `x` and `y` arguments. Each of those options are controlled by a simple argument to `xyPlot`. The plot is filled black circles. The `Plot` argument is a list that controls what is actually plotted, any individual component of that list can be omitted and the default value will be used. This example shows several options. Each line is commented to describe the desired change. they would not necessarily need to be commented in any call.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device would be setPage("square")
> setSweave("graph02", 6 ,6)
> #
> xyPlot(X, Y,
+ # Change from solod black circles to blue plus signs
+ Plot=list(symbol="+", color="blue"),
+ # Set the x-axis range to -2 to 2, for symmetry
+ xaxis.range=c(-2,2),
+ # label at the five integral values: -2, -1, 0, 1, 2
+ xlabels=5,
+ # Change the x- and y-axis titles and end the call
+ xtitle="Random Data",
+ ytitle="Correlated Data")
> # Required call to close PDF output graphics
> graphics.off()
```

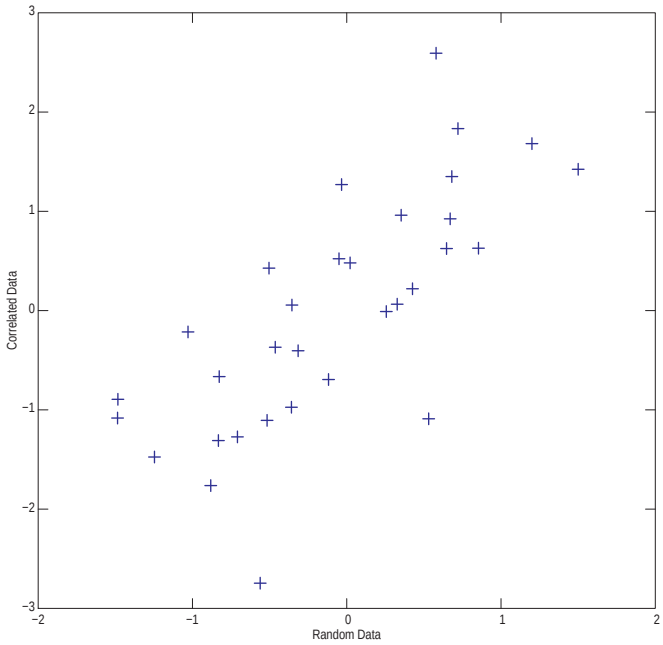

Figure 2. Scatter plot variations.

3 Adding a plot to a graph

All of the high-level plotting functions in the USGSwsGraphs package return much information about the graph and the plot. This information can be used to add a plot to an existing graph or add an explanation.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device would be setPage("square")
> setSweave("graph03", 6 ,6)
> # Create a scatter plot from the X and Y data. The name of the output (AA.pl)
> #  is completely arbiutrary, but consistently used through these examples.
> AA.pl <- xyPlot(X, Y, Plot=list(name="Correlated Data", color="blue"),
+ xaxis.range=c(-2,2), xlabels=5,
+ xtitle="Random Data",
+ ytitle="Response Data")
> # Use the addXY function to add a plot to the graph. The output contains
> #  information about both plots and cen be used to create an explanation
> #  or legend.
> AA.pl <- addXY(X, Z, Plot=list(name="Uncorrelated Data", what="points",
+   color="darkred"), current=AA.pl)
> # The addExplanation function processes the information in the output to
> #  create an explanaiton of the data shown in the plots.
> addExplanation(AA.pl, where="ul", title="")
> # Required call to close PDF output graphics
> graphics.off()
```

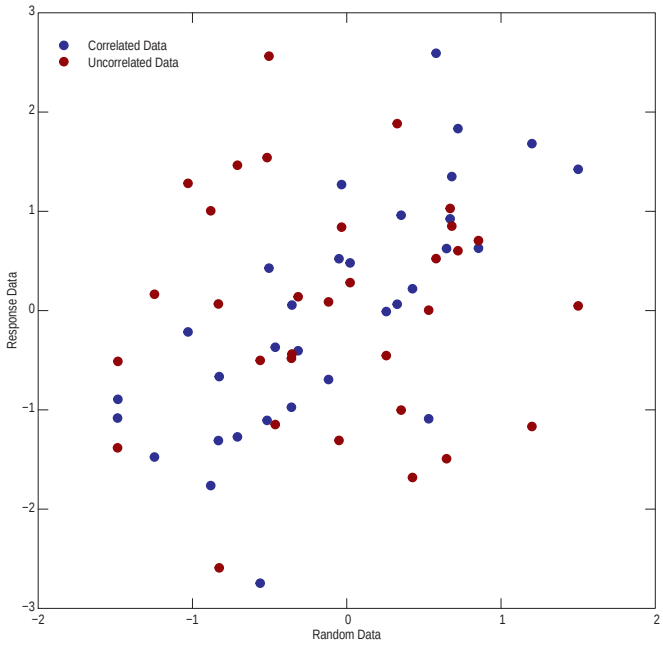

Figure 3. Example of multiple plots in a graph.

4 Multiple Graphs

Sometimes, complete figures need multiple graphs instead of multiple plots within a graph. In that case, or when the graph area needs to be changed from the default, two additional functions (`setLayout` and `setGraph`) are needed. The `setLayout` function can be used to set up a grid of graphs and the `setGraph` functions sets up each individual graph. The graphs set up by `setLayout` can share axes or stand alone.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device would be setPage(layout=list(width=6, height=4)).
> setSweave("graph04", 6 ,4)
> # Set the layout for 2 graphs in one row. and allocate room at the top for
> #  a graph title
> AA.lo <- setLayout(num.cols=2, xtop=1.2)
> # The first graph is the left-most graph
> AA.gr <- setGraph(1, AA.lo)
> # Create a scatter plot from the X and Y data.
> AA.pl <- xyPlot(X, Y, Plot=list(color="blue"),
+ xaxis.range=c(-2,2), xlabels=5,
+ xtitle="Random Data", ytitle="Correlated Data",
+ margin=AA.gr)
> # Add the title
> addTitle("A")
> # The figure caption should always by the lower-left most graph
> addCaption("Figure 4. Example Graphs.")
> # Subsequnect graphs are placed to the right in each row
> AA.gr <- setGraph(2, AA.lo)
> # Create a scatter plot from the X and Y data.
> AA.pl <- xyPlot(X, Z, Plot=list(color="darkred"),
+ xaxis.range=c(-2,2), xlabels=5,
+ xtitle="Random Data", ytitle="Uncorrelated Data",
+ margin=AA.gr)
> # Add the title
> addTitle("B")
> # Required call to close PDF output graphics
> graphics.off()
```

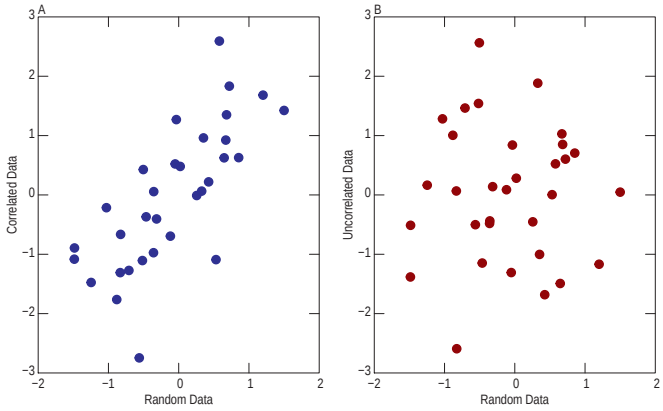

Figure 4. Example Graphs.