hviPlotR: Generating `plot.sas` style figures in R

John Ehrlinger
Cleveland Clinic

Abstract

We introduce the R package **hviPlotR**, a set of tools for creating publication quality graphics in R. The **hviPlotR** package is designed to replace the `plot.sas` macro we currently use in SAS. The package includes both R recipes for generating our standard graphics using **ggplot2** commands and a set of themes designed to format those figures for both manuscript and `PowerPoint` targets.

The goal of this package vignette is to introduce the **hviPlotR** methodology, as well as to document the best practices of creating our publication quality graphics for both manuscripts and power point presentations.

This document is included with the **hviPlotR** package as a package vignette, installed into R when the package is installed, and view able using the `vignette("hviPlotR")` command.

Keywords: publication graphics, powerpoint, ggplot2, plot.sas.

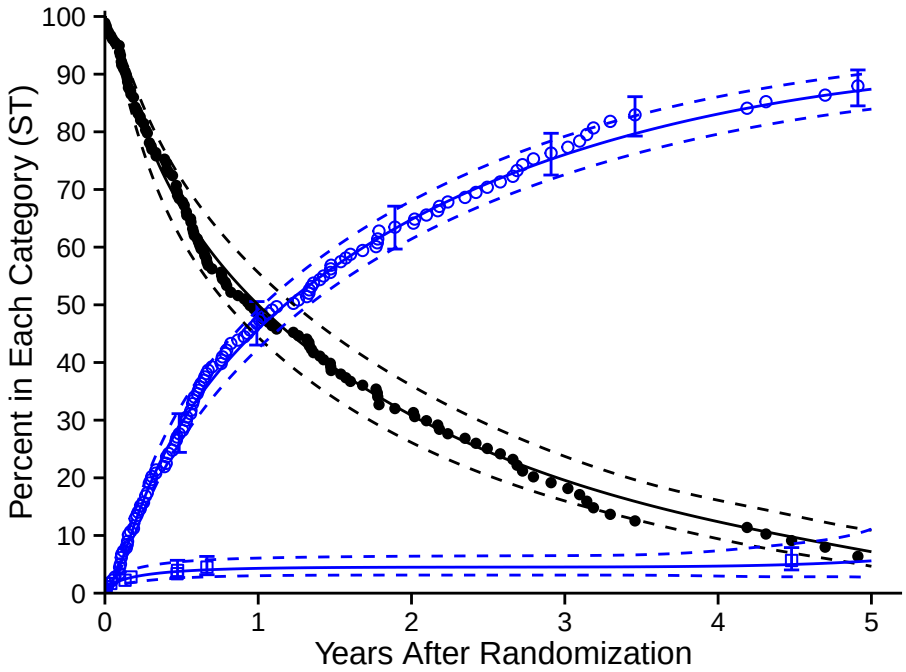

Figure 1: Demonstration figure

1. About this document

This document is an introduction to the R package **hviPlotR**, a set of tools for creating publication quality graphics in R. The package and this document describe the process of creating graphics in R that conform to the standards of the clinical investigations statistics group within The Heart & Vascular Institute at the Cleveland Clinic. These graphics are analogous to those generated with the `plot.sas` macro in SAS.

This document is the package vignette for the **hviPlotR** package, and as such is the primary documentation for the package. The latest version of the document can be obtained with the

```
R> vignette("hviPlotR", package = "hviPlotR")
```

The goal is to update this document as the package is updated to include all relevant changes for publication.

2. Introduction

For many years, the mainstay for generating graphics for manuscripts and presentations in the statistics group in HVI has been the `plot.sas` macro using SAS. However, recently, we have had issues migrating this macro to newer versions of SAS (> 8.0) and MicroSoft Office products (> 2003).

In an effort to alleviate the versifying problems, and to standardize the generation of figures within R, we have developed the **hviPlotR** R package. The goal of the package, and this document, is simplify the creation of publication quality graphics in R. We are specifically encoding the best practices of the HVI Clinical Investigations formatting, so that our statisticians will be able to simply create graphics for publication with a minimal amount of effort.

The **hviPlotR** package also implements best practices for R graphics by leveraging the **ggplot2** package (Wickham 2009). The **ggplot2** package is an implementation of the Grammar of Graphics (Wilkinson 2005), which is a formalization of graphical concepts, and the building of graphical objects from a sequence of independent components. These components can be combined in many different ways.

The `plot.sas` macro is also an implementation of a graphics grammar. The grammar is derived from the ZETA pen plotters, which used GML (Graphics Machine Language) to control between 4 and and 8 colored pens for generating color line and point figures. Because both systems use a graphics language it is straight forward to translate commands between the two methods.

This document outlines how to generate figures using the **ggplot2** and **hviPlotR** packages. Our approach is to demonstrate the R commands to generate the same elements created with `plot.sas` commands. Section 3 gives an overview of the methodology of the `plot.sas` macro and Section 4 details how to create line and point plots with **ggplot2**. A key part of **hviPlotR** package is custom themes for figures. Once you have created your figure, Section 6 details how to get the formatting correct for manuscripts or presentations. Section 7 describes functions for saving the figures to simplify the import into publication documents.

3. The `plot.sas` macro

To demonstrate the process, we first look at some example code using the `plot.sas` macro. This code is intended to generate a figure for manuscript publication and was modified to generate Figure 1. Note the first line of the code block indicates the location of the file.

```
%let STUDY=/studies/cardiac/valves/aortic/replacement/
          partner_publication_office/partner1b/mortality_5y
*******************************************************************************;
* Bring in PostScript plot macro                                               ;
filename plt "!MACROS/plot.sas"; %inc plt;
filename gsasfile pipe 'lp';
*------------------------------------------------------------------------------;
*                                                                              ;
*                      P O S T S C R I P T   P L O T S
*------------------------------------------------------------------------------;
*                                                                              ;
* Multiple decrement, nonparametric and parametric                             ;
filename gsasfile "&STUDY/graphs/ce.states.ST_toJohn.both.ps";
%plot(goptions gsfmode=replace, device=pscolor, gaccess=gsasfile end;
id l="&STUDY/graphs/ce.states.ST_toJohn.sas percent", end;
labelx l="Years After Randomization", end;
axisx order=(0 to 5 by 1), minor=none, end;
labely l="Percent in Each Category (ST)", end;
axisy order=(0 to 100 by 10), minor=none, end;
```

We interupt this command here for some explanation. The `plot.sas` macro call starts with the `%plot` command. The first line sets global graphic values, including the file where the figure will be saved (see Section 7). Each `plot.sas` command is terminated with the `end;` statement. We'll look at each command type individually.

The `id l=` command sets the footnote text used for manuscript figures to identify where the figure is saved (see Section 7). The `labelx` and `labely` commands set the axis label text (Section 4.5) and the `axisx` and `axisy` set the scales for each axis locating text and tics (Section 4.7).

The `tuple` command builds up graphics objects within the figure plot window. The first set of tuple commands builds up a set of three elements containing both points (Section 4.2) and errorbars (Section 4.4). Each `tuple` statement operates on a dataset indicated by the `set` command.

```
/******NON-PARAMETRIC: SYMBOLS AND CONFIDENCE BARS *******/
tuple set=green, symbol=dot, symbsize=1/2, linepe=0, linecl=0,
      ebarsize=3/4, ebar=1,
      x=iv_state, y=sginit, cll=stlinit, clu=stuinit, color=black,
      end;
tuple set=green, symbol=circle, symbsize=1/2, linepe=0, linecl=0,
      ebarsize=3/4, ebar=1,
      x=iv_state, y=sgdead1, cll=stldead1, clu=studead1, color=blue,
      end;
tuple set=green, symbol=square, symbsize=1/2, linepe=0, linecl=0,
      ebarsize=3/4, ebar=1,
      x=iv_state, y=sgstrk1, cll=stlstrk1, clu=stustrk1, color=blue,
      end;
```

Symbols shapes and sizes are specified with the `symbol` and `symbsize` commands (Section 4.3).

The second set of `tuple` statements build up a set of three elements containing lines and confidence intervals (Section 4.1).

```
/**********PARAMETRIC : SOLID LINES AND CONFIDENCE INTERVALS**********/
tuple set=all, x=years, y=noinit, cll=clinit, clu=cuinit,
      width=0.5,color=black,
      end;

tuple set=all, x=years, y=nodeath, cll=cldeath, clu=cudeath,
      width=0.5,color=blue,
      end;

tuple set=all, x=years, y=nostrk, cll=clstrk, clu=custrk,
      linecl=2, width=0.5,color=blue,
      end;
);
run;
*******************************************************************************;
```

The `plot.sas` macro code is closed by the ending `);` characters, and SAS is instructed to `run;` the code. Running combines building the figure by combining elements from `label`, `axis` and `tuple` statements and saving it into the file specified by the `gsasfile` variable. The resulting figure is shown in Figure 2.

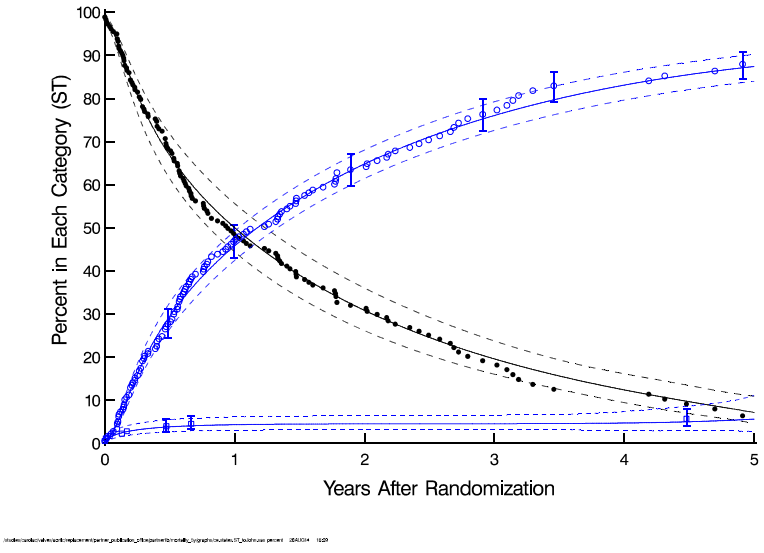

Figure 2: Manuscript figure (SAS version)

Note that much of the figure formatting is mixed within the `tuple` statements using `width`, `color`, `linepe` and `linecl` commands. In the `plot.sas` macro, omitting these commands

will generate a figure with the default values specified within the `device` theme (Section 6). A similar set of `plot.sas` commands is used to create presentation graphics.

```
*-------------------------------------------------------------------------------;
*                                                                               ;
*      C G M   F I L E S   F O R   P O W E R P O I N T   S L I D E S
*-------------------------------------------------------------------------------;
*                                                                               ;
* Competing risks, parametric only                                             ;
filename gsasfile "&STUDY/graphs/ce.states.ST_toJohn.cgm";
%plot(goptions gsfmode=replace, device=cgmmppa, ftext=hwcgm001, end;
axisx order=(0 to 5 by 1), minor=none, value=(height=2.4), end;
axisy order=(0 to 100 by 20), minor=none, value=(height=2.4),
value=(height=2.4 j=r ' ' '20' '40' '60' '80' '100'), end;
tuple set=all, x=years, y=noinit, width=3, color=gray, end;
tuple set=all, x=years, y=nostrk, width=3, color=red, end;
tuple set=all, x=years, y=nodeath, width=3, color=blue, end;
);
run;
```

Differences include the target `device` and `ftext` as well as some handling of figure labels with `value` instead of `label` commands. We also have rules for what to and not to include in presentation graphics (Section 8).

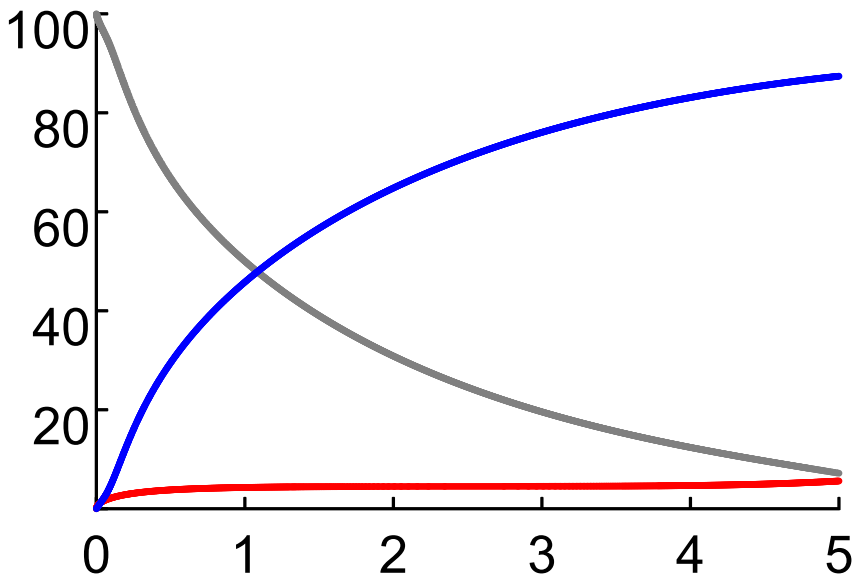

Figure 3: PowerPoint figure (SAS version)

4. Generating ggplot2 graphics

9. Conclusions

In this article, we present some functions in the **hviPlotR** package for R

References

Wickham H (2009). *ggplot2: elegant graphics for data analysis*. Springer New York. ISBN 978-0-387-98140-6.

Wilkinson L (2005). *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. ISBN 0387245448.

Affiliation:

John Ehrlinger
Quantitative Health Sciences
Lerner Research Institute
Cleveland Clinic
9500 Euclid Ave
Cleveland, Ohio 44195
E-mail: john.ehrlinger@gmail.com
URL: http://www.lerner.ccf.org/qhs/people/ehrlinj/
URL: https://github.com/ehrlinger/hviPlotR