

Parallel Coordinates Plot Made Easy

Shane Rosanbalm, Rho, Inc.

ABSTRACT

A parallel coordinates plot is useful for visualizing multivariate data. Unfortunately, there isn't a PARALLEL statement in SGPLOT. In this paper we present a macro called %parallelplot. Using a minimum of parameters (data=, var=, group=) the macro will produce a parallel coordinates plot via SGPLOT.

INTRODUCTION

Producing a parallel coordinates plot in SAS® is not straightforward. The best approach I could find online was from SAS author Prashant Hebbar in his paper from SGF 2012 (Off the Beaten Path: Creating Unusual Graphs with GTL). The outlined process certainly works, but it's written for readability and not for flexibility or scalability. I decided to experiment to see if it was possible to generate a parallel coordinates plot using more flexible and scalable code. The result of this experiment is a macro called %parallelplot which is capable of producing a parallel coordinates plot with a minimum of parameters.

MACRO BASICS

A basic call to %parallelplot looks like this:

```
%parallelplot  
  (data=sashelp.iris  
   ,var=sepalength sepalwidth petallength petalwidth  
   ,group=species  
  );
```

The output produced by the above call is shown in Figure 1:

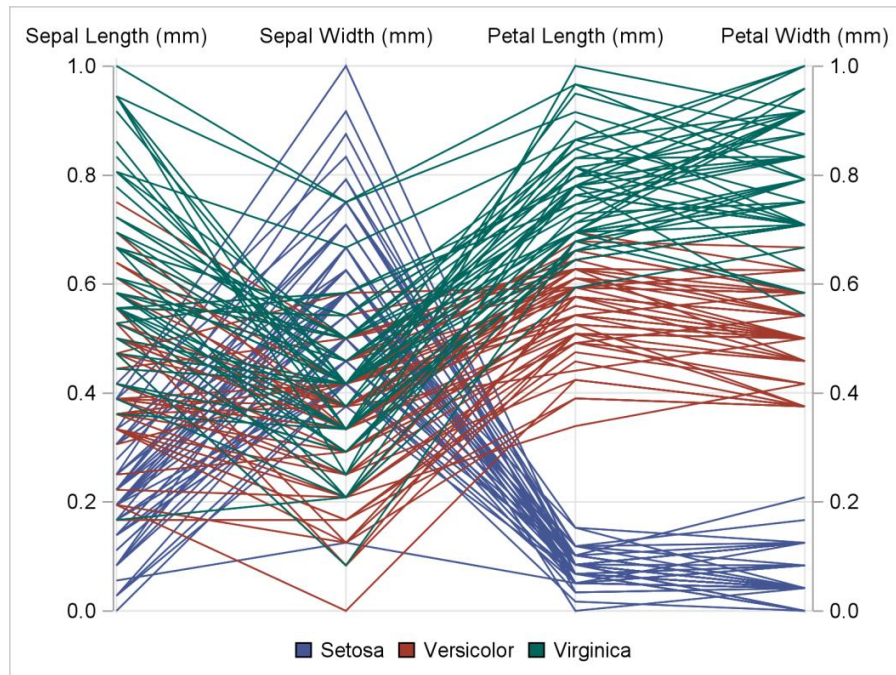


Figure 1. Parallel Coordinates Plot for Fisher's Iris Data

The required macro parameters are `data=` and `var=`, shown in Table 1:

Parameter	Description
<code>data=</code>	Input dataset. Required.
<code>var=</code>	Space-separated list of variables to plot. Required.

Table 1. Required Parameters

Two key optional macro parameters are `group=` and `axistype=`, shown in Table 2:

Parameter	Description
<code>group=</code>	Grouping variable. Optional.
<code>axistype=</code>	Type of y-axis to create. Optional. Valid values: <code>percentiles</code> <code>datavalues</code>

Table 2. Optional Parameters

Specifying `axistype=datavalues` changes the y-axis appearance:

```
%parallel
  (data=sashelp.iris
   ,var=sepalength sepalwidth petallength petalwidth
   ,group=species
   ,axistype=datavalues
  );
```

The output produced by the above call is shown in Figure 2:

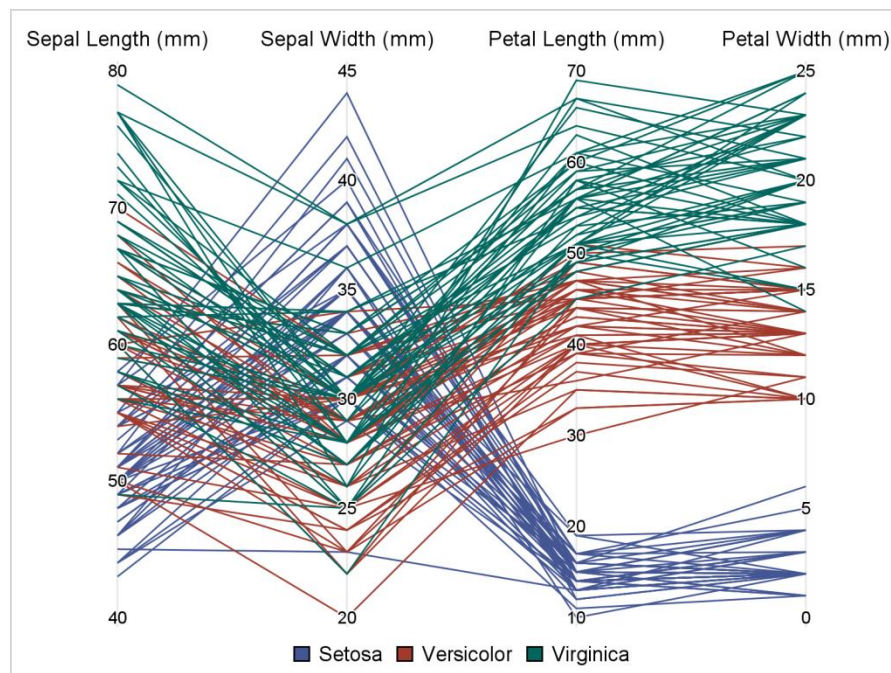


Figure 2. Parallel Coordinates Plot Using Data Values

The `backlight` option is used behind the scenes to put a halo around the tick values, making them (marginally) easier to see against the background of tangled lines.

CUSTOMIZATION

Suppose you want to customize fonts, colors, line patterns, etc. How would you do that? The macro has a fifth parameter, `sgplotout=`, that can be used to capture the underlying SGPLOT code.

Begin by making a "junk" call to the macro. The purpose of this call is *not* to generate output, but rather to capture the SGPLOT code that the macro generates behind the scenes.

```
ods graphics / imagename = "junk";

%parallelplot
  (data=sashelp.iris
   ,var=sepallength sepalwidth petallength petalwidth
   ,group=species
   ,axistype=datavalues
   ,sgplotout=C:/temp/myparallel.sas
  );
```

You then copy the contents of `myparallel.sas` into your main program just below the macro call. In the below example we edit the `styleattrs` of the pasted SGPLOT code. In particular we change the colors of the lines using the `datacontrastcolors=` option.

```
ods graphics / imagename = "junk";

%parallelplot
  (data=sashelp.iris
   ,var=sepallength sepalwidth petallength petalwidth
   ,group=species
   ,axistype=datavalues
   ,sgplotout=C:/temp/myparallel.sas
  );

ods graphics / imagename = "treasure";

proc sgplot data=_pcp50 nocycleattrs noautolegend noborder;
  styleattrs axisextent=data datacontrastcolors=(orange purple green);
  series x=_pcp_var y=_pcp_yval_pct / group=_pcp_series
    groupplc=species lineattrs=(pattern=solid) x2axis name="series";
  series x=_pcp_var y=_pcp_yval_pct / group=_pcp_series
    groupplc=species lineattrs=(pattern=solid) x2axis y2axis;
  text x=_pcp_xtext y=_pcp_ytext text=_pcp_texttext /
    x2axis backlight=1;
  x2axis type=discrete
    display=(nolabel noline noticks)
    grid;
  yaxis display=none;
  y2axis display=none;
  keylegend "series" / exclude=(" ")
    noborder type=linecolor ;
run;
```

Note that we did *not* remove the junk call to the macro. The macro call is still needed in order to generate the work dataset (`_pcp50`) that is used in the SGPLOT code.

The output produced by the above call is shown in Figure 3:

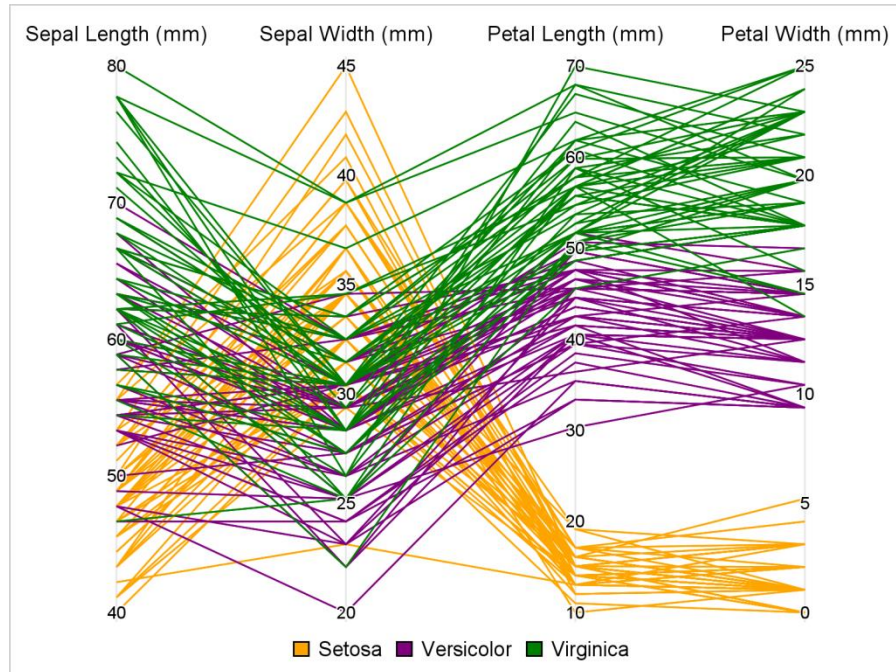


Figure 3. Customization Using SGPLOTOUT=

So pretty!

CONCLUSION

Using the `%parallelplot` macro will greatly reduce the time and effort required to produce a parallel coordinates plot in SAS.

SOURCE CODE

The `%parallelplot` macro is available on GitHub.

<https://github.com/srosanba/sas-parallel-coordinates-plot>

To obtain the macro source code click on the shiny "Clone or Download" button and select "Download ZIP" (shown in Figure 4):

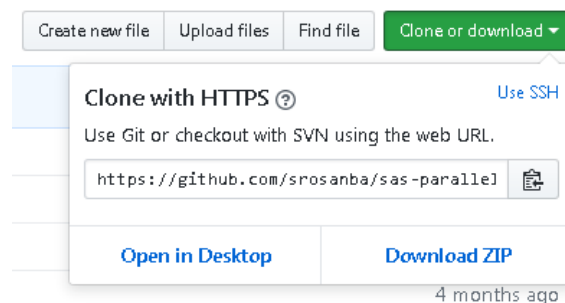


Figure 4. Obtaining Source Code from GitHub

In addition to the macro source code, the ZIP file also contains several example programs to get you up and running more quickly.

Version Requirements: you need 9.4m2 or higher to run this macro.

REFERENCES

Off the Beaten Path: Creating Unusual Graphs with GTL, Prashant Hebbar, SAS Global Forum 2012
<https://support.sas.com/resources/papers/proceedings12/267-2012.pdf>

CONTACT INFORMATION

Your comments and questions are valued and encouraged:

Shane Rosanbalm

Rho, Inc.

Chapel Hill, NC 27516

shane_rosanbalm@rhoworld.com

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.