The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–12

New Graphic Schemes for Stata: plotplain & plottig

Daniel Bischof
University of Zurich
Zurich, CH
bischof@ipz.uzh.ch

Abstract. While Stata's computational capabilities have intensively increased over the last decade, the quality of its default graphic schemes is still a matter of debate amongst users. Clearly some of the arguments speaking against Stata's default graphic design are subject to individual taste, but others are not, such as for instance: horizontal labelling, unnecessary background tinting, missing gridlines, oversized markers. I present two new graphic schemes which attempt to address these concerns. Furthermore, the schemes presented here provide users a set of 21 colors, of which seven colors are distinguishable for people suffering from color blindness. Finally, I give an introduction how users can program their own graphic schemes.

Keywords: st0001, graphic scheme, plotplain, plottig, colorblind

1 Introduction

Stata's computational possibilities have been significantly expanded in the last decade. In contrast, less efforts have been made to extend the graphical possibilities of Stata by providing new graphic schemes. While several small adaptations have been provided both by users as well as the Stata corporation (e.g. Juul 2003), few readymade, uniquely and new graphic schemes have been provided to users. For instance, the recent introduction of William Buchanan's `brewscheme` gives users the opportunity to adapt existing schemes (Buchanan 2016). As such `brewscheme` is a flexible and important advancement of Stata's graphical capabilities. The idea of `brewscheme` is that users can adapt figure designs themselves to fit their demands. While advanced Stata users will find `brewscheme` an excellent facility to adapt Stata's figure design to their likings, many less experienced look for a readymade graphic scheme that looks aesthetically attractive and addresses some concerns – outlined in detail below – in the design of Stata's default graphic schemes.

Thus, in contrast to Buchanan's work, this paper presents entirely new graphic schemes for Stata users. Users then do not need to learn new code and syntax to use these graphic schemes, but can directly employ them after installation. I present two new graphic schemes as well as a short introduction into the code of Stata's graphic schemes to the Stata user community in an effort to make the design of graphic schemes more accessible and potentially attractive to a wider community of users.

The two graphic schemes are specifically designed for users who *a)* are interested in publishing figures in color, *b)* want to make sure that colorblind persons can differentiate colors in graphics, *c)* want to adapt their graphics more closely to suggestions made in Tufte (2001) and Cleveland (1994) and specifically implemented in R's ggplot2 (Wickham 2009). In addition I suggest that these new graphic schemes address some concerns with the default standard graphic schemes. Some of the limitations I raise in this paper rest on personal taste, while other limitations have been raised elsewhere (Juul 2003; Buchanan 2016). The latter issues specifically concern labeling, gridlines, background tinting, framing and the colors employed in figures (Tufte 2001; Cleveland 1994). Stata users oftentimes invest time and effort in writing excessive code to adapt figures (Buchanan 2016). Specifically users who spend time and effort to rearrange figures might rather want to invest their time in writing less code, while thinking about how to present their data in a way that the story of their research stands out.

The first scheme presented here is a minimalistic graphic scheme called `plotplain`. It addresses a range of concerns of Stata's mono graphic scheme (e.g. gridlines parallel to y-axis; horizontal labels on the y-axis; marker size). The second scheme is called `plottig`. It is mainly based on the insights from Cleveland (1994, 100-101) and replicates to a large extent ggplot2 originally designed for the statistical software R (Wickham 2009).

While some of the issues discussed in the upcoming section have already been addressed by Juul's (2003) lean scheme, publication standards have changed significantly since Juul's (2003) publication. Nowadays many journals accept colored figures – especially those published online (e.g. PLOS ONE). Thus, `plotplain` and `plottig` not only present a layout for figures that differs from the lean scheme but also come with two newly composed color palettes. The first palette ensures that each color is distinguishable for colorblind people. Color blindness is not equivalent to a loss of complete color vision. Colorblind people tend to struggle differentiating only certain colors from each other (e.g. red from green). It is a fairly common phenomenon: in a room of 250 people more than 4 % (>ten people) may be colorblind (http://jfly.iam.u-tokyo.ac.jp/color/). We should acknowledge this fact when creating our figures and make sure that we rely on colors which can be distinguished by people who suffer from this condition and use different shapes for each color we draw. The second palette is inspired by colorbrewer (Harrower and Brewer (2011): http://colorbrewer2.org) and provides a palette of 14 colors.

The paper first outlines the key concerns/shortcomings of Stata's default graphic schemes. In a second step I explain how the graphic schemes introduced here attempt to address these concerns. In the final section an introduction is given how users can design their own graphic schemes.

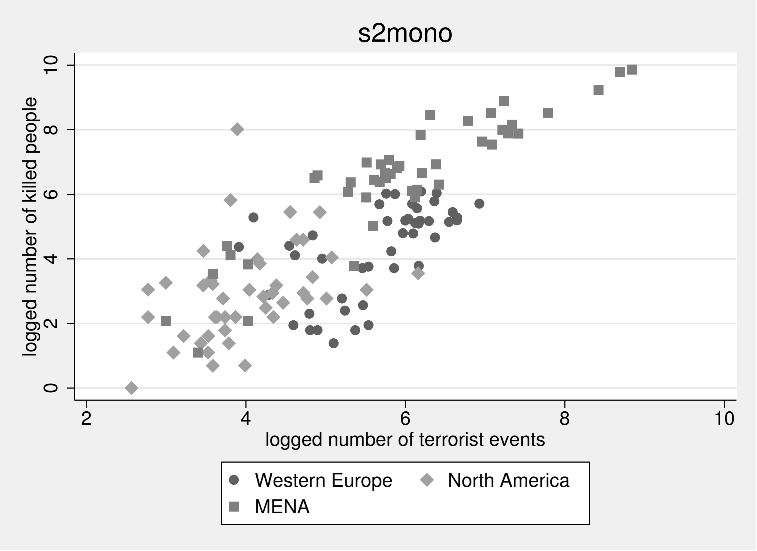
2 Limitations of default Stata graphic schemes

I list all concerns I find with the default Stata schemes below. The critique is mainly based upon Cleveland's (1994) work. Cleveland's major advice and take-away point is: "Make the data stand out. Avoid superfluity" (Cleveland 1994, 24). I suggest that the default Stata figure design often comes with a range of implementations that are superfluous. Thus, the new graphic schemes attempt to avoid such *"chartjunk"* (Tufte 2001). The generic goal of the graphic schemes presented here is to minimize the amount of chartjunk and maximize the information on the data we try to bring across to our audience. In some instances readers might disagree with my opinion. To address these concerns the last part of the paper presents two options to quickly adapt my graphic schemes – both in a newly created scheme as well as with adapting my schemes via Stata code.

Figure 1 shows a scatterplot and figure 2 a lineplot using a frequently employed Stata graphic schemes (s2mono).

```
graph twoway \\\
(scatter nkill nevents if region==12 & nevents<=500 & nkill<=1000) \\\
(scatter nkill nevents if region==7 & nevents<=500 & nkill<=1000) \\\
(scatter nkill nevents if region==6 & nevents>=500 & nkill<=1000), \\\
scheme(s2mono) legend(label(1 Western Europe) \\\
label(2 North America) label(3 MENA)) title("s2mono")
```

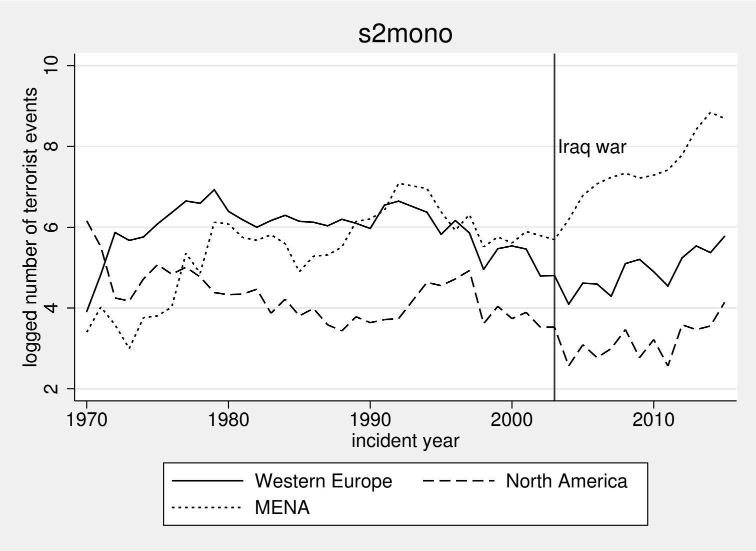
Figure 1: Limitations of s2mono: scatterplot



Source: Author's own.

```
graph twoway \\\
(line nevents iyear if region==12) \\\
(line nevents iyear if region==7) (line nevents iyear if region==6), \\\
scheme(s2mono) \\\
legend(order(1 "Western Europe" 2 "North America" 3 "MENA")) \\\
title("s2mono") xline(2003) \\\
text(8 2003 "Iraq war", place(e)) xlabel(1970(10)2015)
```

Figure 2: Limitations of s2mono: lineplot



Source: Author's own.

The graphics in figures 1 and 2 also outline the major concerns I want to raise and which give reason for improvement:

1. The y-axis labels are vertical which makes reading of the axis inconvenient. Some scholars suggest that vertical labelling saves space. Yet, it also makes the reading of the y-axis more cumbersome. Thus, many Stata users might rather consume more space while simplifying the readability of the y-axis.[1]

2. The unnecessary background tinting (in grey), which also results in the plotarea being surrounded by a frame.

[1] Some users might suggest that vertical labelling makes sense in cases users attempt to bring together several graphics into a single figure. Clearly this is a valid argument. Yet, in most instances users will not attempt to bring together several graphics in a single figure. Thus, I suggest that as a default using horizontal labels constitutes a *status quo* which most users will prefer.

3. Markers and Lines are often too thick. Thus, points in the scatterplot can end up being undistinguishable not because the location of the points are equal but because the markers overlap due to their size. This especially pertains to scatterplots. In figure 1 several data points overlap due to their size, making it difficult for the reader to judge the location of each data point (see: Robbins 2012b, 63 & 165; Cleveland 1994, 24-26). This issue is made worth by an increase of subgroups in a plot.

4. For the same reasons the symbols in the legend could be smaller, otherwise they occasionally overlap.

5. Gridlines are only drawn parallel to the x-axis. Gridlines are more important than a first intuitive reasoning about them might suggest. In fact, human beings are limited in their selective attention. Our judgement of a position of a point or line on one dimension might lead us to misjudge its position on a different dimension (Cleveland 1994). Gridlines can support people to overcome this issue and are also prominently used in Robbins (2012b) and Tufte (2006). They are especially helpful if we want to judge whether two lines correlate (are parallel). Thus, it seems more reasonable that gridlines are drawn parallel to both the x- and y-axes (Robbins 2012a). As suggested in Tufte (2001) gridlines could use a less dominant shape, such as thin dots in order to not distract readers.[2]

6. Overall, most colors used in Stata's default graphic schemes are not distinguishable for colorblind people. Smoothing the difference and fading the colors slightly should enhance readability.

7. There is no obvious necessity to put a frame around the legend.

8. Legends should be placed closer to the data information. Placing legends beneath the figure forces the reader to pay attention to the area beneath the axis (Cleveland 1994, 100). Thus, to ease interpretation a legend could be better placed closer to the data information.[3]

9. Axes use the same color as anything plotted in black. It would be nicer if the axis were more in the background and distinguishable from data points. Albeit, this is a very pedantic point and users might disagree on its usefulness. I will return to the colors of the axes later in the manuscript.

In summary, it appears that a lot of chartjunk could be removed from Stata figures. As suggested earlier Stata users might want to proceed and solve some of these concerns

[2] Some users might disagree on the usage of gridlines parallel to the x-axis. These might be useful if we plot our data using a scatterplot, but they are less useful once we plot time-series data with a lineplot. To address this concern, the final section provides code to get rid of the gridlines in case we would want to plot e.g. time-series data.

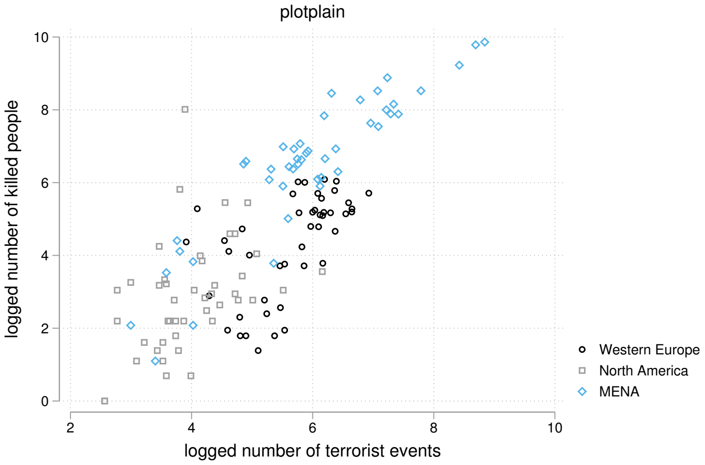
[3] Note that the legend should also rarely be placed within the plotarea. It is only a reasonable approach if the grapher can ensure that such a placement does not distract from the information of the data. Thus, Cleveland (1994, 100) notes: "[a]void putting notes, keys, and markers in the data region".

via syntax or use `brewscheme` to define their own designs. Yet, using graphic schemes that already address the points raised above might be a more efficient and attractive option for most users.

3 plotplain & plottig

The schemes introduced here provide a graphical design that aims to address the key points discussed above. Figures 3 and 4 plot the exact same data and the same Stata code outlind above, but use the scheme `plotplain`. Figure 5 then uses the `plottig` scheme. The former belongs to the s1 scheme family, the latter to the s2 family.

Figure 3: Advantages of plotplain: scatterplot

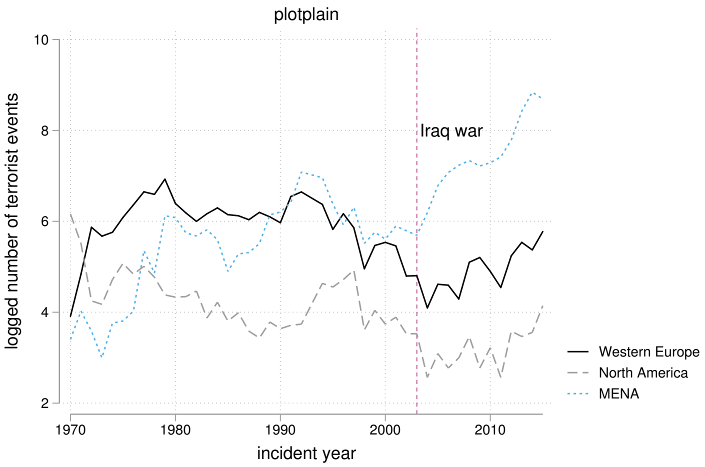


Source: Author's own.

The graphic schemes contain the following improvements:

1. Labels are all horizontal.
2. Markers and lines are thinner. One can clearly tell where data points overlap and where they do not overlap.
3. Legends are placed right of the plotarea and no longer beneath the plotarea.
4. Symbols in the legend are smaller.
5. Frames demarcating legends are omitted.
6. There is no background color.

Figure 4: Advantages of plotplain: lineplot



Source: Author's own.

7. Axes are put into a gray color scheme putting them more into the background of the plot.

8. Gridlines exist for both axes and are marked by thin, gray dots.

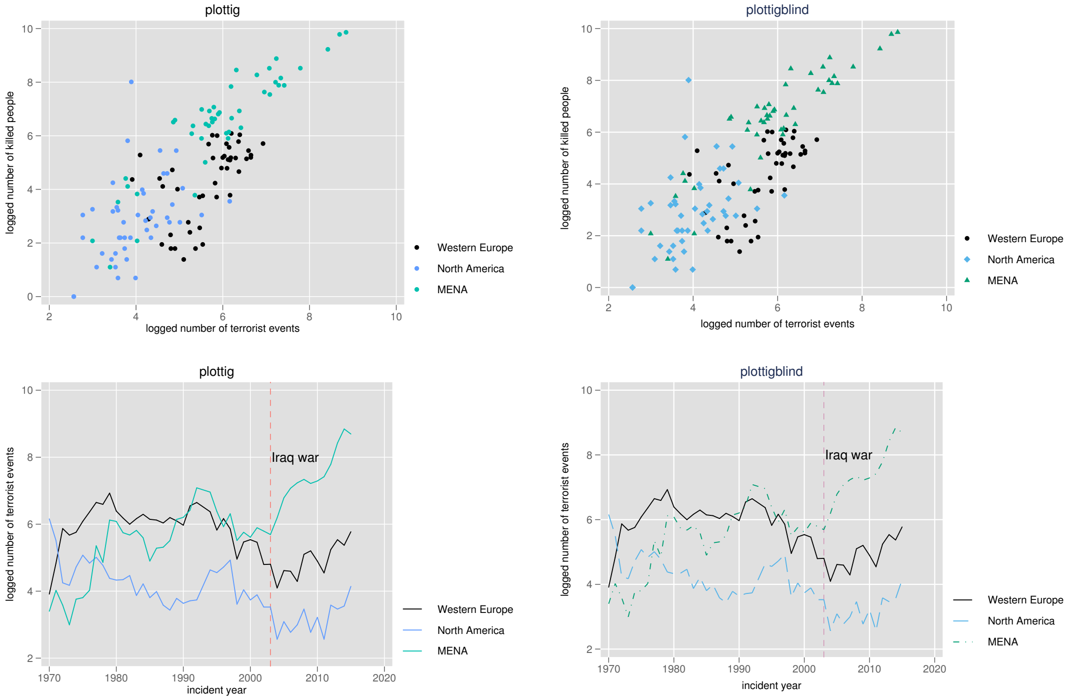
9. Plotareas are not surrounded by any frames.

Both schemes are available in a colorblind friendly version. While `plotplain` relies only on black-white shading and different shapes of markers and lines, `plotplainblind` adds seven colors to the scheme – all distinguishable for colorblind people. Another advantage of the `plotplainblind` scheme is that colors are only drawn if users specify more than two subgroups to be plotted. Single graphs are still black and graphs running across two subgroups are differentiated by black and gray. Any amount of subgroups beyond two are then plotted in seven colors which can be distinguished by colorblind people.

In contrast to `plotplain`, `plottig` explicitly rests on the idea that using a colored background within the plotarea contributes to the legibility of figures. As can be seen in figure 5 this decision enhances the quality of the figures extensively. Markers and especially colors show a higher legibility than it is the case for the `plotplain` figures. Thus, the colored plotarea is not chartjunk but emphasizes the differences between markers and colors. Furthermore, `plottig` makes a new color palette available using 14 colors, while `plottigblind` employs the exact same colors used in `plotplainblind`.

Similar to the lean scheme, both schemes also assign more space to the plotarea than

Figure 5: plottig & plottigblind



Source: Author's own.

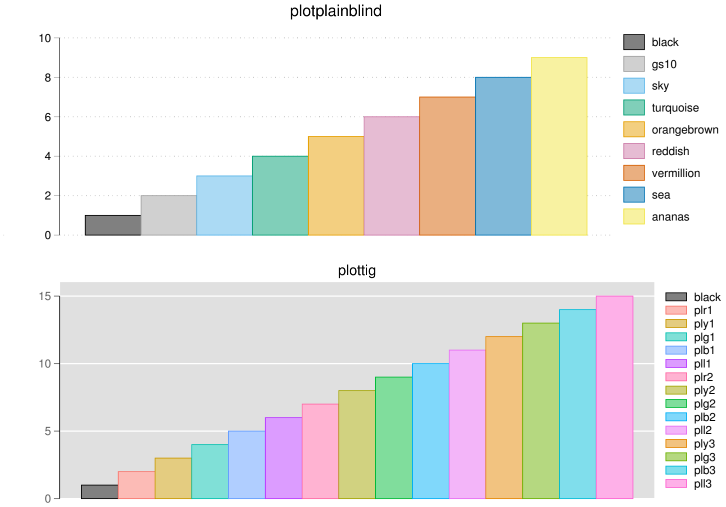
the default Stata schemes (Juul 2003, 300). This is the case because of the adjusted placement of the legend. Again, it ensures that Stata users will maxmize the information on their data in figures and minimize the presence and protrusion of chartjunk.

Figure 6 plots all of these colors using a barplot. Once users installed the schemes, all colors are available in Stata. Colors can be used in syntax commands just like any other color (e.g. `lcolor(vermillion)`) and they might be used in the graph editor by selecting the relevant color from the color palette. The command names of all colors are shown as labels in figure 6.

To use the graphic schemes presented here, users might want to download the files and draw a first line graph using the color "vermillion" by typing in Stata:

```
ssc install blindschemes
set scheme plottig[, permanently]
```

Figure 6: The colors of `plotplainblind` & `plottig`



Source: Author's own.

```
twoway (line x y[, lcolor(vermillion)])
```

3.1 Customizing schemes

A major concern of some advanced users is the adaption or creation of own graphic schemes such as the ones presented in this paper. Besides some information provided in Mitchell (2012, 379-382), not much guidance on how to design a graphic scheme from scratch or how to amend an existing one is available.

The programming language of schemes does not follow the standard Stata syntax. I present some of the key characteristics of a scheme below. First, creation of schemes can be undertaken by using a simple texteditor (.txt file). Yet, users have to make sure that the suffix of their scheme files needs to be .scheme instead of .txt. Commands within a scheme always follow a three step logic:

```
characteristic object option
```

A general characteristic (such as the color) of a respective object (such as the background) is assigned a specific option (such as white):

```
color background  white
```

The example above is taken from the `plottig` scheme and assigns a white background to the scheme. Thus, if we wanted to change the background color we would just need to exchange white in the code above with e.g. black. Below is an overview of the most important commands used within the plottig scheme:

- clockdir = controls position of an object
- color = controls the color of an object
- linepattern = controls the style of a line to be drawn in a figure (e.g. shortdash)
- linestyle = controls the style of a line to be drawn outside of a figure (e.g. shortdash for an axis)
- linewidth = controls the size of a line
- gsize = controls the size of an object
- margin = controls the margins between objects
- numticks_g = controls the number of ticks in an axis
- symbol = the style of a symbol (e.g. circle for scatterplots)
- symbolsize = controls the size of a symbol
- tickstyle = controls the style of ticks (e.g. ticks for labels)
- yesno options = certain options are controlled by simply adding yes or no after a command (e.g. yesno draw_major_grid yes = gridlines for `plottig` scheme)

In general, I would recommend every user interested in creating their own graphic scheme to get started with an existing graphic scheme. This way users can see the actual code and output of the scheme before adapting it to their liking. For instance, some users might entirely disagree with my suggestion to use gray color for the axes in the `plotplain` scheme. To change the colors of the axes users can simply copy-paste the original `plotplain` scheme file which they find in their Stata applications directory. Then they can open a copy of the scheme and search for the following command:

```
color axisline     gs10
```

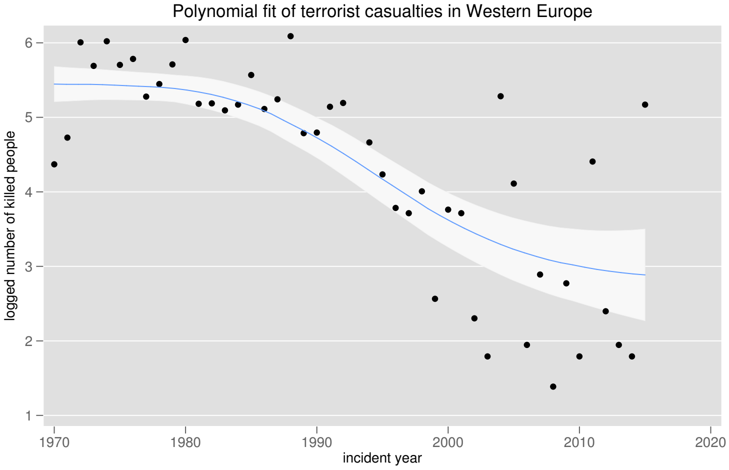
Then users might want to change gs10 into black by deleting the gs10 and writing instead:

```
color axisline     black
```

After saving the newly created scheme file and reseting the general scheme option to use their own scheme, all axes should appear in black. Finally, users might still want to adapt the graphic schemes presented here by using code. For instance, users might want to fade out gridlines entirely. Even though I do not recommend to get rid of gridlines entirely in all applications, there are reasons for excluding them in some applications. While I regard them as a must in scatterplots to assure readers' orientation and especially to assure a simple comparison between the position of data points, gridlines might be less useful if a single time trend is plotted across time. Thus, if users want to use the `plottig` scheme and prohibit the appearance of grids, they might want to use the following command:

```
lpoly nkill iyear if region==12, \\\
title("Polynomial fit of terrorist casualties in Western Europe") \\\
note("") xlabel(, nogrid) ci legend(off)
```

Figure 7: Getting rid of grids



Source: Author's own.

As becomes visible in figure 7 by specifying `xlabel(,nogrid)` users omit the gridlines parallel to the y-axis (more detail on gridlines might be found in: Cox 2009).

Of course this short application just scratches the surface of possibilities. As presented by the graphic schemes introduced here a range of options and changes can be implemented by adapting already existing graphic schemes.

4 Conclusion

This paper attempts to present two new figure schemes to the Stata user community. In doing so it attempts to provide users not only a set of new alternatives to produce figures in Stata but also thereby tries to address some concerns/shortcoming flagged for Stata's default graphic schemes (e.g. horizontal labelling; background tinting; size of markers; placement & style of legends; missing gridlines; colors not distinguishable for colorblind people). Finally, I try to provide users with an introduction how to design graphic schemes for Stata.

A final disclaimer is in order: I do not believe that the graphic schemes provided here are perfect for each user and each application. Some of the concerns are raised are also subject to individual test, while others are not. I hope that the introduction provided with the article guides and motivates users to design their own figure schemes and to share them with the Stata user community.

5 References

Buchanan, W. R. 2016. The brewscheme toolkit for Data Visualization in Stata 1–48. https://wbuchanan.github.io/brewscheme/brewscheme.pdf.

Cleveland, W. S. 1994. *The Elements of Graphing Data*. Monterey, CA: Wadsworth.

Cox, N. J. 2009. Stata tip 82: Grounds for grids on graphs. *Stata Journal* 9(4): 648–651. http://ideas.repec.org/a/tsj/stataj/v7y2007i4p465-506.html.

Harrower, M., and C. A. Brewer. 2011. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal* 7041(July): 261–268.

Juul, S. 2003. Lean Mainstream Schemes for Stata 8 Graphics. *The Stata Journal* 3(3): 295–301.

Mitchell, M. N. 2012. *A Visual Guide to Stata Graphics*. College Station, Texas: Stata Press.

Robbins, N. B. 2012a. Are Grid Lines Useful or Chartjunk? http://www.forbes.com/sites/naomirobbins/2012/02/22/are-grid-lines-useful-or-chartjunk/#10d77b30223f.

———. 2012b. *Creating More Effective Graphs*. Hoboken, New Jersey: Wiley-Interscience.

Tufte, E. R. 2001. *The Visual Display of Quantiative Information*. 2nd ed. Cheshire, Conneticut: Graphics Press.

———. 2006. *Beautiful Evidence*. Cheshire, Conneticut: Graphics Press LLC.

Wickham, H. 2009. *ggplot2: Elegant Graphics for Data Analysis*. Houston: Springer.

About the author

Daniel Bischof is a Senior Researcher (Oberassistent) at the University of Zurich (CH). His main research interests lies in Comparative Politics. He studies party competition, government formation, political representation, public opinion formation and the causes and consequences of political upheaval.