Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements

Brewing color schemes in Stata: Making it easier for end users to customize Stata graphs

Stata Conference 2015
Columbus, OH

Billy Buchanan

Strategic Data Project Data Fellow
Mississippi Department of Education

July 31, 2015

Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements

1. Background & Motivation
    - Stata Graphs
2. Introducing `brewscheme` package
    - `brewscheme`
    - `brewmeta`
3. Other Examples/Use Cases
4. Known Limitations
    - Recycling colors
    - Color Saturations
5. Where to go from Here
    - Currently in the works
    - Future hopes/requests
6. Acknowledgements

Why create a package to create scheme files?

- A non-trivial amount of time in educational research and data analysis (particularly in State/Local Education Agencies) is spent developing data visualizations
- While I'm all for Exploratory Data Analyses (Tukey, 1977), developing production quality graphs customized for the needs of varying stakeholder groups can be time consuming and may required a lot of trial and error to fine tune aesthetics
- Stata's scheme files are a great way to exploit standardization quickly and efficiently, *but* there aren't too many examples of user-written schemes and the few that exist are static and don't allow the end user to tune any of the colors used
- At the same time, tools like Tableau, R, Qlikview, and Trifacta have leveraged new and existing cognitive research to direct the development of their data visualization capabilities in the context of sensitivity to visual perception and/or user interactivity

Where to get the package

- There is a version of the package available on SSC which needs to be updated
- To get the newest version of the package use:
- `net inst brewscheme, from(“http://www.paces-consulting.org/stata”)`
- The package is currently available on GitHub.
- `git clone https://github.com/wbuchanan/brewscheme.g` `master` is most similar to the version available from SSC Archives.
- `git clone https://github.com/wbuchanan/brewscheme.g` `dev` is the most currently updated version (which generally means you wouldn’t want to use it unless you were planning on contributing to the project).

The Good

- Mitchell (2012a) *A visual guide to Stata graphics* is an invaluable resource for customizing Stata graphs and understanding how the optional arguments function across different graph types
- Mitchell (2012b) *Interpreting and visualizing regression models using Stata* is a more focused approach to the use of Stata's graphics for communicating the meaning of statistical models effectively
- Cox (2014) *Speaking Stata graphics* is a compilation of Stata Journal articles/columns with additional helpful tips on efficiently creating and using Stata's graphics capabilities

The Bad

Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements

- But not everyone is so happy about Stata's graphics (particularly with s2color)
  - "I'd been wondering for a while how to avoid the default blue background in my graphs without having to write the specific option every single time." (Anonymous blog comment on BuRd scheme)
  - "I do not like the default stata figure schemes and many of my colleagues do not like them either. So far I handled this problem by typing exhausting syntaxes in order to work around the issues involved in stata's default figure schemes." (Daniel Bischof's Blog (FEB2015))
  - "Stata graphs are not the nicest part of the software. What Stata wins on making it possible to recode or regress a set of variables in one line, it loses when it comes to making the look of a plot a bit cleaner or simply more elegant. Stata graph syntax is rather usable, but the default schemes are rarely satisfactory." (SRQM Blog Post)
- **BUT** it doesn't have to be this way…

The Bad

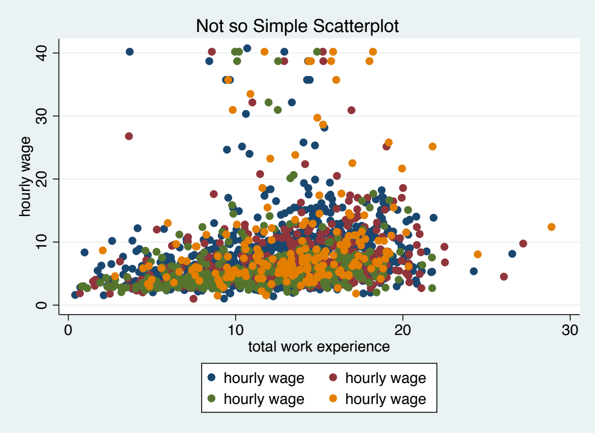
(continued)

So, what is it exactly that these users are complaining about?

```
sysuse nlsw88.dta, clear
tw scatter wage ttl_exp if industry == 11 || scatter wage
ttl_exp if industry == 4 || scatter wage ttl_exp if
industry == 6 || scatter wage ttl_exp if industry == 7,
ti("Not so Simple Scatterplot", c(black) span
size(medlarge))
```

The Bad
(continued)

While Stata provides end-users with a fair amount of flexibility (*less alpha transparency*), with great flexibility comes great responsibility for users not to create graphs that look like:



The Ugly

Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements

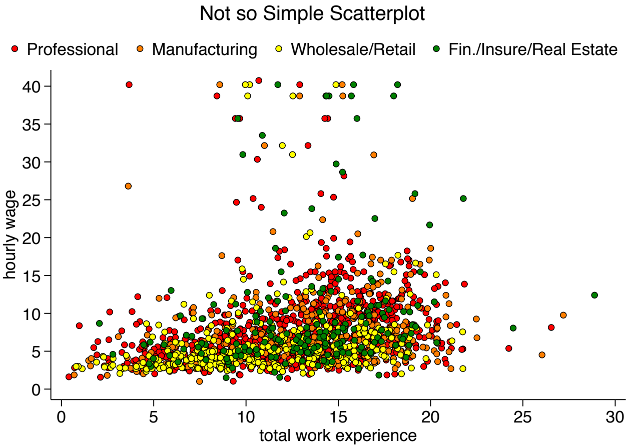
So, we end up trying to code around the defaults provided by the default scheme loaded by Stata and end up with code that looks something like:

```
tw scatter wage ttl_exp if industry == 11, mc(red)
mlc(black) mlw(vthin) msize(medsmall) || scatter
wage ttl_exp if industry == 4, mc(orange) mlc(black)
mlw(vthin) msize(medsmall) || scatter wage ttl_exp
if industry == 6, mc(yellow) mlc(black) mlw(vthin)
msize(medsmall) || scatter wage ttl_exp if industry
== 7, mc(green) mlc(black) mlw(vthin)
msize(medsmall) ylab(#10, angle(0) nogrid)
legend(pos(12) span label(1 "Professional") label(2
"Manufacturing") label(3 "Wholesale/Retail") label(4
"Fin./Insure/Real Estate") rows(1) region(lc(white)
fc(white) ic(white)) size(medsmall) symy(2.5)
symx(2.5)) graphr(ic(white) fc(white) lc(white))
plotr(ic(white) fc(white) lc(white)) xlab(#7)
ti("Not so Simple Scatterplot", c(black) span
size(medlarge))
```

The Ugly
(continued)

Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements

Just to get something that looks like:



Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements

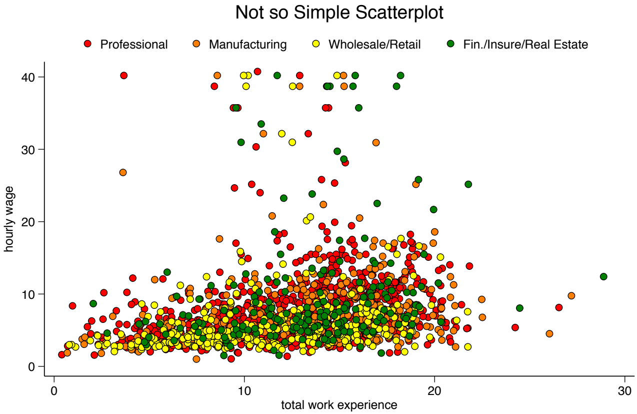
One palette to rule them all

Creating a scheme with a single color palette for all graph types

- If a single set of colors is all that is needed, we can create a scheme file fairly quickly/easily:
- `brewscheme, scheme(onePaletteToRuleThemAll) allst(mdebar) allc(5)`
- Then the ugly syntax graph is reduced to something like:

```
tw scatter wage ttl_exp if industry == 11 || scatter wage
ttl_exp if industry == 4 || scatter wage ttl_exp if
industry == 6 || scatter wage ttl_exp if industry == 7,
ti("Not so Simple Scatterplot")
scheme(onePaletteToRuleThemAll) legend(label(1
"Professional") label(2 "Manufacturing") label(3
"Wholesale/Retail") label(4 "Fin./Insure/Real Estate"))
```

One palette to rule them all



One palette to rule them all

Adding additional overlaid plots with only a single color palette

- Sometimes, you may want to include other plots that help to summarize the patterns in the data

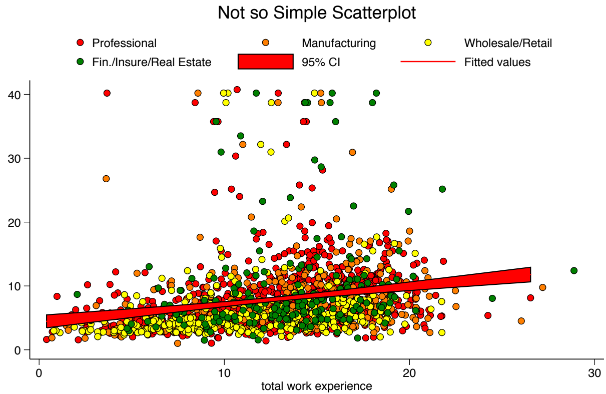
```
tw scatter wage ttl_exp if industry == 11 || scatter wage
ttl_exp if industry == 4 || scatter wage ttl_exp if
industry == 6 || scatter wage ttl_exp if industry == 7 ||
lfitci wage ttl_exp if industry == 11, ti("Not so Simple
Scatterplot") scheme(oneSchemeToRuleThemAll) legend(label(1
"Professional") label(2 "Manufacturing") label(3
"Wholesale/Retail") label(4 "Fin./Insure/Real Estate")
rows(2))
```

One palette to rule them all

With multiple plot types

Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements

But, given the similarity in colors this is neither the most aesthetically pleasing nor the most effective method to communicate the relationships we are trying to visualize:

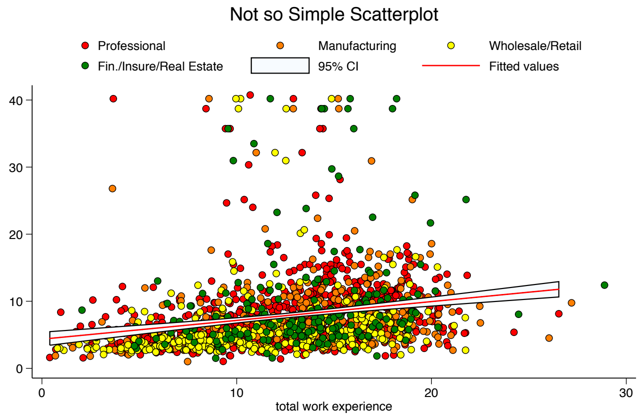


One palette to rule all the rest

- In the case above, we want to use some other color palette for lines/confidence intervals to distinguish them from the scatterplot points...
- `brewscheme` makes the creation of these types of schemes easily as well
-
  ```
  brewscheme, scheme(onePaletteToRuleTheRest1)
  somest(mdebar) somec(5) linest(pastel) linec(5)
  cist(blues) cic(5)
  ```
- Or, we can change the color saturation from the original example
-
  ```
  brewscheme, scheme(onePaletteToRuleTheRest2)
  somest(mdebar) somec(5) somesat(10) linest(mdebar)
  linec(5) linesat(30) cist(mdebar) cic(5) cisat(80)
  ```

One palette to rule the rest

Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements



One palette per graph type

Holy bat colors Batman

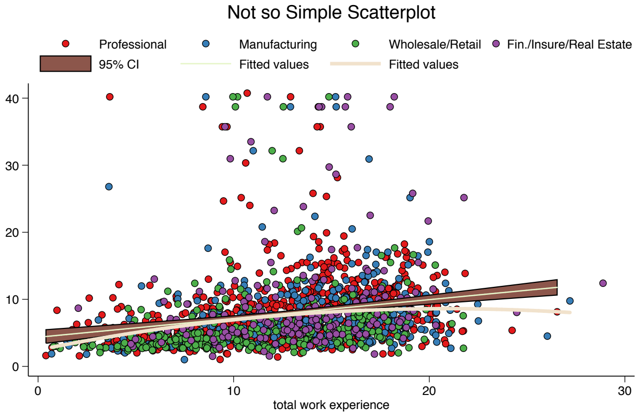
- If you really wanted, you could specify a different color palette for every graph type parameter of the scheme files

```
brewscheme, scheme(myriadColorPalettes) barst(paired)
barc(12) dotst(prgn) dotc(7) scatstyle(set1) scatc(9)
linest(pastel2) linec(8) boxstyle(accent) boxc(8)
areast(dark2) areac(8) piest(mdepoint) sunst(greys)
histst(veggiese) cist(activitiesa) matst(spectral)
reflst(purd) refmst(set3) const(ylgn) cone(puor)
```

One palette per graph type

Holy bat colors Batman

Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements



What are the properties of the palette I am using?

Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements

- What if you want to know if the combination of your selected color palette and number of colors is color blind friendly?
  - `brewmeta pastel2, colors(5) prop("colorblind")`
  - `The color palette pastel2 with 5 colors is Not color blind friendly`
  - The results on the bullet above get printed to the results window
  - And are also returned in a local macro r(pastel25_colorblind)
- You can also use this function to find out information about "friendliness" in the context of LCD viewing, Color printing, and Photocopying.

What are the properties of the palette I am using?

- `brewmeta paired, colors(9)`

```
The color palette paired with 9 colors is Not color blind
friendly
The color palette paired with 9 colors is Possibly LCD
friendly
The color palette paired with 9 colors is Not photocopy
friendly
The color palette paired with 9 colors is Possibly print
friendly
```

College-Going Readiness Toolkit

(the program cgwaterfall is a program used as part of a presentation at SDP's 2015 Convening)

- Generate the scheme used for replicating/improving graphs used as part of the Strategic Data Project's (SDP) College-Going Readiness Toolkit.
- The data are not real, but the program used to generate the graphs has default options to indicate whether a graph should be publicly released or not.

```
brewscheme, histst(pastel1) barst(puor) barc(11)
boxst(pastel2) boxc(8) scatst(set1) scatc(9)
somest(oranges) somec(9) cist(dark2) areast(greys)
scheme(sdpdemo)
```

```
cgwaterfall using analysis/CG_analysis, schn(first_hs_name)
bys(sep) year(2005 2005) ont(ontime_grad)
f(enrl_1oct_ninth_yr1_any) s(enrl_1oct_ninth_yr2_any)
grad(chrt_grad) coh(chrt_ninth) si(sid) st(sd mean)
scheme(sdpdemo) sex(male)
```

MISSISSIPPI DEPARTMENT OF EDUCATION

Ensuring a bright future for every child

College-Going Readiness Toolkit

On-Track Status Visualization

Background & Motivation

Stata Graphs

Introducing brewscheme package

brewscheme

brewmeta

Other Examples/Use Cases

Known Limitations

Recycling colors

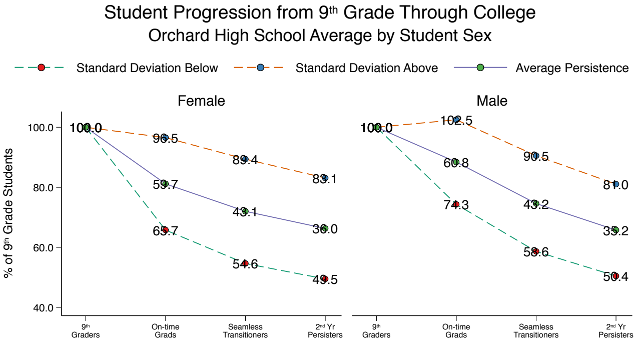
Color Saturations

Where to go from Here

Currently in the works

Future hopes/requests

Acknowledgements



Sample: 2004 through 2005 first-time ninth graders. Students who transferred into or out of are excluded from the sample.
Postsecondary enrollment outcomes from NSC matched records. All other data are from administrative records.

May include PII.
Not for Public Release

Graphing Categorical Data

Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements

- Data are from syslog records of educators' use of a student assessment platform and included timestamps and report types accessed over the span of several months
- We fitted a multi-level latent class model to estimate within user latent classes (e.g., how would we classify the way educator X used the data system during a given instance) and between user latent classes (e.g., is educator X a "power user", a "sporadic" user, etc ...)
- So how would we visualize the relationship between the proportion of reports accessed and between user latent classes to illustrate some type of profile of platform use for each class?

Between User Classes by Report Type

(Thanks to Nick Cox for `spineplot`)

Created a custom graph scheme to use for entire project:

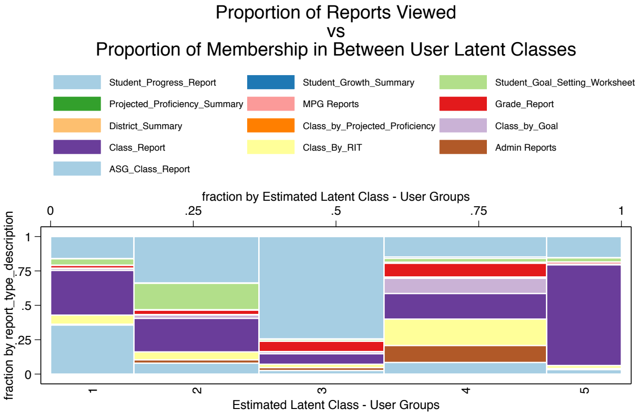
```
brewscheme, scheme(sdpcapstone) barstyle(paired) barc(12)
dotstyle(prgn) dotc(7) scatstyle(set1) scatc(9)
linest(dark2) linec(8) somestyle(accent) somec(8)
areast(dark2) areac(8)
```

Added the newly created scheme to the options of `spineplot (available from the SSC archives):`

```
spineplot reportcat ugroups, xlab(, labsize(small)
angle(90) axis(2)) scheme(sdpcapstone) legend(cols(3)
rows(5) pos(12) size(vsmall)) ti(“Proportion of Reports
Viewed” “vs” “Proportion of Membership in Between User
Latent Classes”)
```

Between User Classes by Report Type

(Thanks to Nick Cox for spineplot)



Only a single parameter to set the recycling of colors

- Because the `pcycle` parameter only takes a single value, the colors options for each graph types don't always work exactly as intended
- It'd be great if there were independent pcycle options for each graph type to allow the colors to be recycled independently for each graph type
- Barring that happening, it means heading back to the drawing board to replicate similar behavior based on the maximum value of the number of color arguments

Adjusting the saturation/intensity of the colors

- I found when testing examples for the slides recently, that the saturation arguments weren't being correctly interpretted in the scheme file
- `help scheme entries` and `scheme-s2color.scheme` don't provide the clearest guidance regarding how to solve this, but maybe some of the folks from StataCorp could help with this?
- While alpha transparency would likely provide greater utility (e.g., helps with overplotting, etc ...), Stata graphics do not include any alpha transparency/blending methods at this point
- Following the conference, I was able to test the program a bit more extensively and it seems that I had picked a poor example, because it was working but can sometimes be much more subtle than anticipated.

brewcolors

proofing tools

- Martin Krzywinski's website includes several text files with RGB interpolations to simulate
  - Deuteranopia — color blindness sensitive to Green's
  - Protanopia — color blindness sensitive to Red's
  - Tritanopia — color blindness sensitive to Blue's
- The hope is that the function will essentially "replay" the graph for all - or individual - types of colorblindness, translate the RGB values using the lookup tables on Krzywinski's website, and use `graph combine` to create a single "proof" image allowing users to test how others with any of the types of colorblindness would see the graph

brewcolors

adding colors to Stata

- An XKCD survey of 222,500 user sessions resulted in a list of 900+ “named” colors.
- Thanks to an old Statalist (listserv) message from Vince Wiggins responding to a question from Bill Rising from 2004, the way to define named colors for Stata is fairly well documented and easy to follow.
- It isn’t likely that many people would want to install all of the colors, so it would probably be helpful to add a feature to display colors before installing them — sort of a search functionality.

brewpalette

function to add palettes to existing brewscheme/brewmeta files

- Currently, the palettes are predefined/hardcoded with validation checks performed against the named/known palettes.
- Would like to add a function allowing users to define their own palettes with an interface that might look like:
- `brewpalette “0 0 255” “0 255 0” “255 0 0” “0 0 0” “255 255 255”, name(mypalette)`
- This would add to a Stata data file that serves as a look up table for the colors and their properties (e.g., colorblindness, print, photocopy, and LCD friendliness).

With some help from some Java Savvy friends

Background & Motivation
Stata Graphs
Introducing brewscheme package
brewscheme
brewmeta
Other Examples/Use Cases
Known Limitations
Recycling colors
Color Saturations
Where to go from Here
Currently in the works
Future hopes/requests
Acknowledgements

- It'd be very interesting to see how a Stata integration with the Prefuse Java library would look/work
- Prefuse was a precursor to Protovis, which eventually birthed the D3 javascript library
- Prefuse was developed to be interactive, which could help Stata facilitate more robust/thorough exploratory data analyses (Tukey, 1977).

Acknowledgements

- The individuals and organizations below were instrumental in supporting the development of `brewscheme`, provided valuable/useful feedback around the work motivation the package along the way, and/or allowed me to take a bit of time away from the office to be here today
- There are a few folks that deserve acknowledgement for their help, support, and/or feedback:
  - Patrick Ross, Chief School Performance Officer, Mississippi Department of Education (MDE)
  - Paula Vanderford, Executive Director of Accreditation and Accountability, MDE
  - Staci Curry, Director of Accountability Support, MDE
  - Tollie Thigpen, Director of Accountability Systems, MDE
  - The Strategic Data Project at the Center for Education Policy Research at Harvard University
  - Eric Moore, Director of Research, Evaluation, and Accountability, Minneapolis Public Schools
- Hopefully, this helps others to reduce their workflows and create more uniformly styled graphs and if anyone has any suggestions or wants to collaborate feel free to let me know Billy Buchanan