ggconventions: Easy graphical conventions in ggplot2

Trevor L Davis

Abstract—There are a number of cultural conventions that could be used in a graphic, such as using blue for males and red/pink for females. If such a convention is strong enough one could use it in graphics to improve comprehension by reducing the cognitive load of the viewer (less need to refer to the legend) as well as make a more visually appealing graphic. Combining multiple conventions (using a male/female icon in addition to color) could reduce the cognitive load through a redundancy gain. We develop an R package that provides a framework for an individual or organization to develop their own conventions and functions to easily add graphical conventions to ggplot2 graphics, one of the more popular graphics frameworks in the mainstream graphical (and statistical) toolkit R.

Index Terms—graphical conventions, grammar of graphics, ggplot2

1 INTRODUCTION

There are many cultural conventions that associate colors, shapes, and other features to concepts that can be used to facilitate graphical understanding by increasing the salience of graphed objects and reduce the need to consult a legend. For example one can use the color pink or red for Female and the color blue for Males. In finance one often uses green or black to represent profits or positive balances and red for losses and negative balances (hence the term “in the black” and “in the red”). One frequently uses the color blue to represent water and in certain graphs the color green is used to represent parks. One could also use shapes such as the sign ♀ for female and ♂ for male or use an icon of a woman in a skirt and a man in pants. Certain maps frequently use symbols such as an oil pump or a chaff of wheat to represent centers of resource production. By using colors, shapes, and other visual clues (perhaps description or choice of scale in the axis or other context) the graphic designer may make a more visually appealing graphic and perhaps more importantly by using salient conventions may reduce the cognitive load of the viewer making it more easier to understand the graphic (by perhaps reducing the need to flip back and forth to the legend).

We implement a framework for easily adding conventions to R ggplot2 graphs. It would allow an individual or organization to develop their own conventions and provides utility functions for using those conventions in ggplot2 graphs. The framework is developed as an R package. You can install the development version right now with the following R commands:

```r
if(require("devtools")) install.packages("devtools") devtools::install_bitbucket("ggconventions", user="trevorld")
```

Eventually it will be released on the Comprehensive Repository Archive Network (CRAN).

---

1. Red seems to be the more popular choice over pink in graphics out in the wild for female, see for example http://thesocietypages.org/graphicsociology/tag/gender/
2. For example see http://commons.wikimedia.org/wiki/File:Stavangerbykart.png
3. For a critical discussion of such a convention see http://thesocietypages.org/graphicsociology/2012/08/21/global-smoking-rates-by-gender/
The `ggconventions` package provides a framework for an individual or institution to define a number of graphical conventions (described in section 4) and some functions to make it easier to add these graphical conventions to one's `ggplot2` plot (described in section 5). For example here is the code to make a normal `ggplot2` graph:

```r
library("ggplot2")
data("gender_data", package="ggconventions")
ggplot(gender_data,
aes(x=class, y=survivors, colour=gender, shape=gender)) + geom_point(size=7) + ggtitle("# adult survivors of Titanic")
```

And here is the code to make a conventions aware `ggplot2` graph using `ggconventions` and a predefined gender convention:

```r
source(system.file("conventions/gender.R", package="ggconventions"))
ct_ggplot(gender_data,
  aes(x=class, y=survivors, colour=gender, shape=gender)) + geom_point(size=7)
```

Notice that the only difference after loading in the graphical convention was the use of `ct_ggplot` instead of `ggplot`. One can even set up configurable conventions that can be modified by calls to the `options` function:

```r
options(gender_shape = "bathroom")
ct_ggplot(gender_data, aes(x=class, y=survivors, colour=gender, shape=gender)) + geom_point(size=7)
```

```r
options(gender_shape = "emoji")
ct_ggplot(gender_data, aes(x=class, y=survivors, colour=gender, shape=gender)) + geom_point(size=7)
```

### 2 RELATED WORK

There is a large body of work in the study of graphics on picking effective colors, shapes, and other graphical attributes. For example Cleveland et al [3], Tufte [7], Heer et al [4][9], Durand [5], Fairchild [6], Brewer [8], and Gooch et al [10]. Much of the research has focused on the good choice of color palettes and shapes that offer good performance in terms of perception, discrimination, and salience and the interactions between shapes and colors. There does not seem to have been much research on how the choice of colors and shapes based on cultural conventions increases salience for individuals from that culture (and perhaps decreases salience for individual outside it). Although we do not directly perform such research, by providing a framework for conventions...
we can make it easier for researchers to find and run experiments on various conventions.

On the other hand, in practice in the real world there have been many graphics that make use of various cultural conventions. For example the footnotes in the introduction had links to a number of graphs that used gender conventions. Many maps use symbols such as that of an airplane to represent an airport[^4^] and use the color blue to represent bodies of water. A number of domain specific tools have built-in conventions - for example Open Street lets you build maps with some pre-defined map icons[^5^].

Many finance applications will automatically use red and black to display financial balances. However most general purpose graphical systems, although sensibly choosing color palettes for generic categorical and continuous variables, do not detect and adjust for pre-existing cultural conventions. Although they usually let you manually specify your colors and other graphical elements, not many graphical systems let you easily create your own conventions. We step in and provide a framework for individuals and organizations to create their own graphical conventions for use with the ggplot2 graphical library in the R graphical (and statistical) toolkit. In ggplot2 conventions are expressed through the use of various "scales", so we provide a way to have your custom scales be called with minimal syntactic sugar.

### 3 Adding a Convention Using GGConvention Framework

In order to create your own convention you must define a function that converts your raw data into a "convention" class. An example of a gender convention is below. It must be named as_convention_MYCONVENTION where you replace MYCONVENTION with the name of the variable you would like to apply the convention to. If many variable names have similar values you can simply assign your original function to new function of a different name. The example convention uses a sex convention to define a gender convention. This function should clean up the data as desired, assign it the "convention" class, and assign it a "convention" attribute with its name (that will be used to call scales).

```r
# contents of gender.R from before require("ggconventions")
# create gender convention function
as_convention_gender <- function(x) {
  y = factor(toupper(x), levels=c("MALE", "FEMALE"))
  class(y) <- c("convention", class(y))
  attr(y, "convention") <- "sex"
  y
}

as_convention_gender <- as_convention_x
```

You must also define scales to be used with your convention. We continue the gender example below. At the minimum you should specify the scales you anticipate using, in the gender example that would be a color and shape scale, but since it is hard to anticipate what your user will call it doesn’t hurt to define a scale for all possible dimensions, in our gender example we simply fall back on the default categorical scales for the x and y axis. Common ggplot2 scales are x, y, colour, fill, shape, and linetype.

```r
# contents of gender.R continued
# create scales for our conventions
scale_colour_sex <- function(...) {
  scale_colour_manual(...,
    values=c("FEMALE"="deeppink1", "MALE"="blue"))
}

scale_fill_sex <- function(...) {
  scale_fill_manual(...,
    values=c("FEMALE"="deeppink1", "MALE"="blue"))
}

scale_shape_sex <- function(...) {
  convention <- getOption("gender_shape", "default")
  switch(convention,
    bathroom = scale_shape_manual(...,
      values=c("FEMALE"="U0001F6B9", "MALE"="U0001F6B8")),
    emoji = scale_shape_manual(...,
      values=c("FEMALE"="♀", "MALE"="♂")),
    scale_shape_manual(...,
      values=c("FEMALE"="♀", "MALE"="♂"))
  )
}

scale_x_sexx <- function(...) {scale_x_discrete(...)}

scale_y_sexx <- function(...) {scale_y_discrete(...)}

scale_linetype_sexx <- function(...) {
  scale_linetype_discrete(...)
}
```

### 4 The ggconventions API

For full details of the API enter help(package="ggconventions") in an R prompt. Here is a high level overview of all the functions provided by ggconventions. They provide lots of syntactic sugar for adding conventions to one's graphs with minimal changes to underlying ggplot2 code assuming one has loaded in a convention.

#### 4.1 c_transform and c_fortify

The c_transform and c_fortify functions take a data frame and transform the columns (by column name) according to any conventions we have defined. c_fortify adds the new transformed to the data frame (by default prefixing "c." to the name whereas c_transform replaced the columns with the new convention altered data. When using these transformed data frames with ggplot2 we will automatically call the scales we have defined in our convention so we can directly use these modified data frames with functions like qplot and ggplot. Here is an example of using qplot with c_transform:

5. [GitHub's map icons](https://github.com/openstreetmap/map-icons)
And here is an example of using ggplot with c_fortify. We most often want to use c_fortify over c_transform when we want to plot our original data in one dimension (in the example on the y-axis) and use our transformed data along another axis (in our example color).

```r
source(system.file("conventions/redblack.R", package="ggconventions"))
data("profit_data", package="ggconventions")
ggplot(x=Index, y=profit, colour=c.profit, shape=c.profit, data=c_fortify(profit_data)) + geom_bar(stat='identity')
```

4.3 load_conventions and list_conventions

The functions load_conventions and list_conventions allows one to load and list the convention files contained in a package. By default they list and load the conventions contained in the ggconventions package but when new packages contain their own conventions this can be used to load one or all of them. If you are developer creating your own package that will include graphical conventions you may place them wherever you like but we’d recommend that you place them in inst/conventions (which after installation will be placed in the conventions directory).
folder of your package which is the default for list_conventions and load_conventions).

```r
list_conventions(package="ggconventions")
```

```r
## [1] "gender.R" "redblack.R"
```

```r
# just one convention
load_conventions("redblack",
    package="ggconventions")
```

```r
# all of the conventions in the package
load_conventions(package="ggconventions")
```

### 4.4 add_c_scales

Unfortunately at the present time the internals of ggplot2 are such that our convention scales are automatically called with normal ggplot2 graphs but they are not automatically called if we specify multiple facets. add_c_scales adds back these scales for simple plots with multiple facets. It only supports the most basic aes() calls so it is recommended to do all your data transformations in the data first and then call ggplot. Here is an example of its use:

```r
require("ggplot2")
load_convention("gender", "ggconventions")
data("gender_data", package="ggconventions")
gg <- qplot(x=class, y=survivors, colour=gender,
    shape=gender, geom="blank",
    data=c_transform(gender_data)) +
    geom_point(size=5)
```

```r
# The following are equivalent
add_c_scales(gg + facet_grid(. ~ gender))
gg + facet_grid(. ~ gender) +
    scale_colour_gender() + scale_shape_gender()
```

```r
# gg + facet_grid(. ~ gender) # gives an error
```

**REFERENCES**


[6] Landa, Fairchild, “Charting color from the eye of the beholder”.


