A Minimal Demo of `glyphmaps`

Garrett Grolemund

June 4, 2012

This document demonstrates the basic features of the `glyphmaps` package (name may change). `glyphmaps` was written to create plots like the ones below, which recreate plots similar to those in Wickham (2011) and Wickham et al. (Submitted). Each plot is made with `glyphmaps`. To run the examples in this document please download the `glyphmaps` package with the code below. (Note: Install won't work until I finish documenting the namespace. But cloning and load$_all()$*DOESwork*.)

```
library(devtools)
install_github("ggplyr", "garrettgman", "glyphmaps")

## Installing github repo(s) ggplyr/glyphmaps from garrettgman
## Installing ggplyr.zip from https://github.com/garrettgman/ggplyr/zipball
## Installing glyphmaps
## Error: Command failed (1)

library(glyphmaps)

## Error: there is no package called 'glyphmaps'
```

```
ggplot(seasons) + map_us + grid(geom_line(aes(x = time, y = pred)), grid.aes = aes(lon,
    lat), x.nbin = 58, y.nbin = 25, ref = ref_box(aes(fill = avg), alpha = 0.2, color =
NA))

## Error: could not find function "ggplot"

ggplot(nasa) + map_nasa + glyph(geom_star(aes(r = ozone, angle = date, x = 0, y = 0,
    fill = mean(temperature))), major.aes = aes(long[1], lat[1]), glyph.by = c("long",
"lat"))

## Error: could not find function "ggplot"
```

1 Glyphs

The common feature of these graphs is that they are built around glyphs. Glyphs are geometric objects (i.e, geoms) designed to display information within the geom. In other words, a glyph can display information even if it is drawn by itself, without references to an external coordinate system. In reality, all geoms are a type of glyph, but the term glyph is usually reserved for complicated geoms, such as those that contain their own internal coordinate systems. The star glyphs in Figure **??** illustrate how glyphs can contain an internal (minor) coordinate system and can still be plotted in an external (major) coordinate system.

Glyphs reveal a hierarchical structure of graphics: every plot is a collection of geoms, each of which can be thought of as its own self contained plot. Sometimes these subplots are not very interesting, as in the subplot created by a single point geom. At other times, thee subplots are quite complex, as in the star glyph of Figure **??**.

Graphs inherit this hierarchical structure from the data they describe. Data is produced through an iterative process of collecting observations, grouping observations and summarizing groups of observations to create more compact, information dense sets of data. Humans innately perform this process when collecting data; it is a cognitive pattern of the human brain which I will write about in the second cognitive chapter of my thesis.

Glyph maps simultaneously expose data from multiple levels of the hierarchy. As a set of geoms, the glyphs reveal relationships between data points in the higher level, compact data set. As individual plots, each glyph retains information about the data points in the lower level group of data that it summarises. This dual display makes glyph maps particularly useful for certain data analysis tasks. It also provides two different approaches to constructing glyph maps.

Glyph maps can be built from the top down by treating each glyph as a geom within the plot of interest. ggplyr provides new geoms $\texttt{geom}_s tar, \texttt{geom}_r adar, \texttt{geom}_d art, and \texttt{geom}_p lyr which allow a user to quickly incorporate geoms into an exis$

```
Error in paste(repo, branch, sep = "/", collapse = ", ") :
  argument "repo" is missing, with no default
```

.

2 top down methods

ggplyr's top down methods help users fit glyphs into existing methods of visualization. These methods provide new geoms (often based on new grobs) that can be used alongside existing geoms in ggplot2.

Top down methods are difficult to implement in ggplot2 for two reasons

First, ggplot2 calculates aesthetics on the entire data set at once. However, glyphs contain aesthetics that must be keyed to subsets of the data.

Second, the final width and height of individual glyphs often depends on non-position aesthetics, such as angle and length. In the ggplot2 pipeline, these aesthetics are scaled right before the plot ranges are trained. As a result, the final widths of glyphs must be computed at draw time and frequently place parts of the glyph outside the plot window.

2.1 $\textbf{geom}_p lyr$

ggplyr provides a way for users to compute aesthetics by

The first element of x is

```
Error in eval(expr, envir, enclos) : object 'x' not found
```

. Boring boxplots and histograms recorded by the PDF device:

```
## two plots side by side (option fig.show='hold')
par(mar = c(4, 4, 0.1, 0.1), cex.lab = 0.95, cex.axis = 0.9, mgp = c(2, 0.7, 0),
    tcl = -0.3, las = 1)
boxplot(x)

## Error: object 'x' not found

hist(x, main = "")

## Error: object 'x' not found
```

Do the above chunks work? You should be able to compile the TEX document and get a PDF file like this one: `https://github.com/downloads/yihui/knitr/knitr-minimal.pdf`. The Rnw source of this document is at `https://github.com/yihui/knitr/blob/master/inst/examples/knitr-minimal.Rnw`.

References

Hadley Wickham. The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, 40(1): 1âĂŞ29, 2011. URL `http://www.jstatsoft.org/v40/i01/`.

Hadley Wickham, Heike Hofmann, Charlotte Wickham, and Diane Cook. Glyph-maps for visually exploring temporal patterns in climate data and models. *Environmetrics*, Submitted.