sugarglider: Create glyphmaps of spatio-temporal data

by Maliny Po and Dianne Cook

Abstract (An abstract of less than 150 words.)

1 Introduction

`Note: use similar terminology as cubble & glyph-maps. Add a quick start (quick guide on how to use the sugarglider`

2 Literature Review

Glyph Maps

`(Desperately need more resources)`

- Glyph maps are a specific type of multivariate glyph plots where each spatial location is represented by a glyph that encapsulates data measured over time at that location. As detailed in Hadley Wickham's paper, glyph maps are particularly effective for uncovering both local and global structures, emphasizing temporal relationships within the data. These maps utilize small glyphs or icons to represent multiple data values at each location, extending the concept of glyphs which are traditionally used to display multivariate data.
- Challenges with Faceted Maps and Spatio-Temporal Animations: While faceted maps and spatio-temporal animations are useful for highlighting spatial patterns, they often fall short in adequately showcasing temporal trends. To overcome this, a transformation technique is employed which utilizes principles of linear algebra to convert temporal coordinates (minor coordinates) into spatial coordinates (major coordinates). This transformation is implemented in packages such as GGally and cubble, facilitating a more integrated approach to spatio-temporal data visualization.
- The R package cubble introduces an innovative cubble class designed to efficiently organize spatial and temporal variables. This dual structure allows for separate or combined manipulation of these variables while maintaining synchronization. A spatial cubble object is constructed from distinct spatial and temporal tables through the function make_cubble(), requiring the specification of three attributes: key, index, and coords. This functionality not only simplifies the data wrangling process but also enhances the analytical capabilities when dealing with complex datasets.

Extending ggplot2 with ggproto

`Diversify your resources a bit :((`

- Elegant Graphics for Data Analysis: The architecture of ggplot2 is fundamentally based on the ggproto system of object-oriented programming. Initially, ggplot2 utilized the proto system, developed by Grothendieck, Kates, and Petzoldt in 2016, for object-oriented tasks. This system, described in detail in the Proto package documentation, outlines that proto is an S3 subclass of the R environment class, implying single inheritance and mutable state characteristic of all environments. Proto objects are created and modified using the proto function which sets the parent environment, evaluates expressions, and handles lazy evaluation of arguments.

However, as the need for an official extension mechanism in ggplot2 grew, the limitations of the proto system became apparent, leading to the adoption of ggproto. This transition is well-documented in Hadley Wickham's book, ggplot2: Elegant Graphics for Data Analysis, which also introduces how to utilize ggproto objects to extend ggplot2 functionalities.

The creation of a new ggproto object is facilitated by the ggproto() function, which requires the name of the new class and an existing ggproto object from which it will inherit. For instance, to introduce a new statistical transformation, one might create a ggproto that inherits from `Stat` and `Geom`. However, merely creating a ggproto object does not make it accessible or useful to the end user.

(Example from ggplot2-book.org)

```
NewObject <- ggproto(
  `_class` = NULL,
  `_inherits` = NULL
)
```

To bridge this gap, the creation of a layer function is necessary. An example is the `new_stat()` function, which follows a consistent format: setting defaults in the function arguments, and calling layer(), which handles the distribution of these arguments into either geom parameters, stat parameters, or aesthetics. This function exemplifies the methodology to create functional and user-accessible components in ggplot2.

(Example from ggplot2-book.org)

```
new_stat <- function(mapping = NULL, data = NULL,
                     geom = "geometry", position = "identity",
                     na.rm = FALSE, show.legend = NA,
                     inherit.aes = TRUE, ...) {
  layer(
    stat = NewStat,
    data = data,
    mapping = mapping,
    geom = geom,
    position = position,
    show.legend = show.legend,
    inherit.aes = inherit.aes,
    params = list(na.rm = na.rm, ...)
  )
}
```

While developing ggplot2 extensions, it may seem intuitive to encapsulate extensions as new geoms, as they are frequently used by users to add layers to a plot. However, the diversity in ggplot2's capabilities often stems more from the variety in statistical transformations (stats) than merely geometric objects (geoms), suggesting a nuanced approach in designing extensions that effectively enhance the plotting system.

3 Software

```
Check out:  https://journal.r-project.org/articles/RJ-2023-013/
Note: collect all the technical part in one section
```

Data structure

Rescale

Spatial-temporal transformation

Aesthetics

Parameters

Interactivity

Examples

4 Application

Five examples are selected to demonstrate various features of the sugarglider package: (1) Creating a ribbon glyph map to visualize annual fluctuations in minimum and maximum daily patronage for each train station, revealing seasonal trends. (2) Using glyph segments to compare patronage on typical weekdays versus weekends at different stations, enabling insights for optimizing service schedules. (3) Utilizing glyph ribbons to represent variations in patronage during distinct peak periods—AM peak, interpeak, PM peak, and late PM hours—across stations. (4) Displaying differences in patronage across

transportation modes (Metro, VLine, or both) using glyph segments, identifying capacity imbalances. Lastly, (5) Employing glyph ribbons to compare public and school holiday patronage against regular days, aiding in scheduling and resource planning.

Yearly Patronage Changes by Station

Weekday vs. Weekend Patronage

Patronage During Different Peak Times

Patronage Variations by Transportation Mode

Public and School Holiday Patronage vs. Regular Days

5 Discussion

6 Acknowledgements

7 References

- Glyph-maps: https://vita.had.co.nz/papers/glyph-maps.pdf
- cubble: https://www.jstatsoft.org/article/view/v110i07
- Proto: https://cran.r-project.org/web/packages/proto/proto.pdf
- GGally: https://cran.r-project.org/web/packages/GGally/GGally.pdf
- Elegant Graphics for Data Analysis: https://ggplot2-book.org/

8 References

Maliny Po
Monash University
Department of Econometrics and Business Statistics
Melbourne, Australia
ORCiD: 0009-0008-4686-6631
malinypo12@gmail.com

Dianne Cook
Monash University
Department of Econometrics and Business Statistics
Melbourne, Australia
ORCiD: 0000-0002-3813-7155
dicook@monash.edu