Review of the micromaps R package

Marcus W. Beck

June 23, 2015

To address:

- Do you find the syntax of the micromap package easy to understand?
- Could the main functions of the package, create_map_table, mmplot, and mmgroupedplot, be improved?
- Is there anything about the structure of micromap objects that could be improved for instance, in order to use micromap objects in R Shiny, or to make setting micomap plot options more expressive (such as adding plotting elements to a geom in ggplot2)?
- Is there some functionality of the package that you think is missing?
- Is the documentation in the Introduction Guide, which comes with the package, and the JSS article helpful?

Ideas:

- I strongly encourage use of S3 object methods for defining the functions. This makes it much easier to determine what object classes can be used as input to each function. For example, the `create_map_table` function requires a `SpatialPolygonsDataFrame` object as input but this it not clear from the documentation - all that is specified is 'shapefile'. Defining an S3 method for this class should create a generic and `SpatialPolygonsDataFrame` definition in the help file and will also ensure that a sensible error is returned if an unsupported object class is passed to the function. This style of documentation is really easy using the roxygen and devtools packages. For example...

```
#' This is a function
#'
#' This is some more detail about the function
#'
#' @param input SpatialPolygonsDataFrame input
#' @param ... other arguments
#'
#' @export
```

```r
mmplot <- function(input, ...) UseMethod(mmplot)

#' @rdname mmplot
#'
#' @export
#'
#' @S3method mmplot SpatialPolygonsDataFrame
mmplot.SpatialPolygonsDataFrame <- function(input, ...){

  # function stuff

}
```

- Related to the last point, the documentation could be improved in some places to clarify the exact object class required for the arguments to each function. For example, The help file for `mmplot` indicates that `stat.data` is a 'table' but this really should say 'data.frame'.

- Data input that is linked to the map data could be provided in raw format, with a user-supplied summary function to improve flexibility and reduce burden on the user, e.g., additional arguments passed to mmplot or mmgroupedplot

- This is more of a personal preference, but I'm not too crazy about a new graphics device being created each time a the function is executed. This can lead to rapid cluttering of the desktop.

- The `create_map_table` function could be used internally within `mmplot` to skip a step, i.e., SpatialPolygonsDataFrame could be used as input directly.

- Similarly, polygon simplification could be incorporated into the package

- Seems like a lot of this could be simplified using only the SpatialPolygonsDataFrame as input. The data slot could include the raw data used in the `stat.dat` argument, which can be further summarized for display as a dot plot, boxplot, or other summary.

- Maybe return a warning or error if colors does not equal length of groupings?

- I had to do a bit of searching to find the Information Guide as it's buried in the R library. Maybe supply a link in the description file so it shows up as a link on the CRAN web page.

- Seems like this package could be setup very nicely using an expressive style as in ggplot2 - this might be a lot of work though and probably requires substantial overhaul of package structure.

- aspect ratio of maps with resizing of graphics device

- From JSS manuscript - add histrograms, ts, symbol plots, etc. and incorporate interactivity (via shiny)

- Break up ideas into easy fixes and future features...