**Improving R Interactive Graphics via HTml Package**

Jong Hyun BAE

**Email:** bnbbkr@gmail.com

**Short description:** This project aims to improve the R Interactive Graphics via HTml (RIGHT) package that enables interactive data visualization and analysis to explore data and gain valuable insight. RIGHT is the first package that implements linked graphics using HTML canvas and JavaScript by highlighting the relationship among multiple plots. We plan to improve RIGHT in two ways. One is to execute R code for analysis on a server, overlay the results on the plot, and update them interactively in response to hiding data. For instance, the loess() in R can be called to overlay a local regression line on a scatter plot and updating it as outliers are removed (hidden). The other is to support ggplot2-like R API to create complex visualization more easily. Since HTML5 canvas and JavaScript will be used to create the visualization, it can be delivered to virtually any device/platform with a modern web browser. The improved RIGHT will become a powerful tool for understanding data and delivering analysis results.

**Project title: Improving the R-interactive-Graphics-via-HTml (RIGHT) Package**

**Project short title (30 characters): Improving RIGHT**

URL of project idea page: <http://rwiki.sciviews.org/doku.php?id=developers:projects:gsoc2014:right>

**Abstract**

There is a flood of data generated around us every day from social media, mobile clients, financial transactions, and so on. However, understanding and exploiting “big data” poses many challenges as the opportunities they promise. The value of data can be monetized only when they are properly analyzed, and the role of data visualization is becoming more important for data analysis. As a result, many static visualization systems have been developed, and excellent software packages, including lattice and ggplot2 in R, exist to create them. This project aims to improve the R Interactive Graphics via HTml (RIGHT) package that enables interactive data visualization and analysis to explore data and gain insight. RIGHT is the first package that implements linked graphics using HTML canvas and JavaScript in order to highlight the relationship among multiple plots. I plan to improve RIGHT in two ways. One is to execute R code for analysis on a server, overlay the results on the plot, and update them interactively in response to hiding data. For instance, the loess() in R can be called to overlay a smooth regression line on a scatter plot and update it as outliers are removed (hidden). The other is to support ggplot2-like R API to create complex visualization more easily. Since HTML5 canvas and JavaScript will be used to create the visualization, it can be delivered to virtually any device/platform with a modern web browser. The improved RIGHT will become a powerful tool for understanding data and delivering analysis results.

**CONTACT INFORMATION**

**Student name:**  Jong Hyun BAE

**Link\_id:**  bnbbkr

**Student postal address:** #1901, 533dong Hyosung Apt, Beksul-village, Jeongja-dong, Jangan-gu, Suwon-si, Gyeonggi-do 440-841, Korea.

**Telephone(s):** +82-10-9048-5971

**Email(s):**[bnbbkr (at) gmail (dot) com](mailto:sch8906@gmail.com)

**Student affiliation:**

**Institution:** Sungkyunkwan University (SKKU), Korea

**Program:** Bachelor of Science in Semiconductor Systems Engineering

**Stage of completion:** Graduating in February 2015

**Contact to verify:**

Prof. Jae W. Lee ([jaewlee (at) skku (dot) edu](mailto:jaewlee@skku.edu))

Chung Ha Sung ([sch8906 (at) gmail (dot) com](mailto:flammy@gmail.com))

**Bio of Student:**

My name is Jong Hyun Bae, and I am a senior student in the Department of Semiconductor Systems Engineering at Sungkyunkwan University (SKKU) in Korea. I will receive my bachelor’s degree in February 2015. I am very excited about this application for Google Summer of Code (GSoC). I have strong software development skills with C/C++, Javascript, and R, and am confident that I will make this project a great success. I started working on this project in February 2014, initially as a senior project with the mentors. I have already looked into the current version of RIGHT with my mentors and have plans on what and how to improve the RIGHT package. I like this project, and I believe that many R users will benefit from the RIGHT package as I achieve the goals by the end of summer.

**Schedule Conflicts:**

I may work as a software engineering intern at Samsung electronics for a month in July (not yet confirmed). Even if this happens, I would continue to work on this project regularly in the evening and during the weekends.

**MENTORS**

(This information is available on the project ideas pages)

**Mentor names:**  Prof. Jae W. Lee, Dr. Jung Hoon Lee, Chung Ha Sung

**Mentor emails:** [jaewlee (at) gmail (dot) com](mailto:jaewlee@gmail.com), [flammy (at) gmail (dot) com, sch8906 (at) gmail (dot) com](mailto:flammy@gmail.com)

**Mentor link\_ids:** jaewlee, junghoon, sch8906

**Have you been in touch with the mentors? When and how?**

* I have been working on a pilot project for this proposal with the mentors as a senior project since February 2014. I have held regular meetings every two weeks and many ad-hoc meetings with the mentors both in person and via conference calls.

**CODING PLAN & METHODS**

**Main steps & Planning:**

**This project consists of two main components:**

**1. Adding server offload functionality to enable both client- and server-side interactivity**

* This feature will significantly improve interactivity on weak client devices from server clouds to mobile clients.
* A client-server communication protocol will be defined, and routines to support server-side computing will be implemented.
  + The shiny package uses Websockets to communicate between the server and client. However, the Websockets package is removed from CRAN and new code for communication is necessary.

**2. Adding ggplot2-like R API and readjusting R-JavaScript interface**

* We learned during GSoC 2013 that base graphics-like R API is not suitable to create complex interactive plots, so we are planning to develop a set of R API similar to what ggplot2 offers. We will look into ggvis package to make best use of what is already developed.
* R already has rich functions to design a plot, e.g. scales and labels package. To take advantage of this, we would like to adjust the R-JavaScript interface such that
  + Plot layout is mostly constructed in R; and
  + JavaScript provides low-level plotting functions for drawing, which are called by R.

**Evaluation:**

**Three main aspects will be evaluated.**

1. **Functionality**

* ggplot2-like R API should be supported to easily construct complex plots.
* RIGHT should be able to run R code on the server side to update plots interactively in response to hiding data.
* Chrome, FireFox, and recent versions of IE should be supported without changing any settings.

1. **Performance**

* The performance goal of RIGHT is to handle datasets with 50K+ entries while maintaining reasonable response time.

1. **Ease of Use**

* To ensure ease of use and high quality of documentation, I plan to hire 5 “beta testers” who will test the functionality of RIGHT and provide feedback to improve its usability.

**Testing:**

**Three levels of testing will be performed to ensure** [**completeness of**](http://endic.naver.com/enkrIdiom.nhn?idiomId=9072873317614801b310e4415d57f2e9) **this project:**

1. **Unit test**

* Unit test is very important for a successful open-source project, yet it is well known that it is tricky to apply the unit testing framework to GUI code. Therefore, I will try my best to clearly separate the “code that draws” from the rest, and sufficient unit testing code will be written for the non-drawing routines. I am planning to use Qunit to setup the unit testing framework.

1. **Application level test(UI)**

* Three problems will be used to test the overall functionality of the package.
* Application test 1: 1000 randomly sampled rows from diamond dataset will be used to create 5 types of plots (scatter, histogram, box and whisker, line, and pie) to test layout and user interaction.
* Application test 2: Outlier removal and interactive update will be tested using a linear regression problem.
* Application test 3: Whether the RIGHT package can handle dataset with more than 50K rows will be tested on a web browser (Chrome or FireFox).

1. **User testing**

* The R package will be tested by at least 5 testers in late August and early September to improve usability. Mobile platform, e.g. Android, will be tested to visualize the result.

**Documentation:**

**The final R package will have the following documentations distributed through CRAN .**

* Code-level documentation for R and JavaScript.
  + All the exported R functions will have good documentation with meaningful examples.
  + Key JavaScript functions will be well documented with some examples.
  + Comments will be added to the code.
* Tutorials-level documentation using application.
  + Demo video will be renewed.
  + Tutorials will be provided to help users create various plots using the R API.

**Perceived obstacles and critical objectives:**

* Functionality issues
  + R API to support server offloading for various types of analysis.
  + No more required setting for browser to use.
* Performance issues
  + Handling 50K data sets for all supported plots without significant increase in response time.
  + Server offloading without significantly increasing the response time.

**TIMELINE:**

**The main schedule will be divided by the following three parts.**

* Previous RIGHT cleanup and testing
* Server offloading
* Expand supported types of functions

**Detailed schedule will be following.**

|  |  |  |
| --- | --- | --- |
| **Date** | **What to do** | **GSoC Timeline** |
| Week 1 (5/19~5/26) | - Refactor previous RIGHT and R API. - Begin server offloading part. | 5/19: Students begin coding for their Google Summer of Code projects. |
| Week 2 (5/27~6/03) | - Work server offloading part |  |
| Week 3 (6/04~6/11) | - Implement unit testing for existing code and document the code. |  |
| Week 4 (6/12~6/19) | - Improve server offloading part. |  |
| Week 5 (6/20~6/27) | - Improve server offloading performance. |  |
| Week 6 (6/28~7/05) | - Application level testing about three cases for functional factors( i.e. response time) |  |
| Week 7 (7/06~7/13) | - Testing and refinement for mid-term evaluation.  - Submit mid-term evaluations |  |
| Week 8 (7/14~7/21) | - Testing and refinement for mid-term evaluation. |  |
| Week 9 (7/22~7/29) | - Begin function expanding and improve R API. | 7/23: Mentors and students can begin submitting mid-term evaluations.  7/27: Mid-term evaluations deadline |
| Week 10 (7/30~8/06) | - Function expanding and improve R API and document about tutorial-level using application |  |
| Week 11 (8/07~8/14) | - Refactor code to improve code quality.  - Submit final evaluations |  |
| Week 12 (8/15~8/22) | - User testing and improve R API  - Submit final evaluations | 8/22: Final evaluation deadline and begin submitting required code sampled to Google. |
| Week 13 (8/23~8/25) | - Waiting final results | 8/25: Final results of GSoC 2014 announced. |

**Contingency plan for things not going to schedule:**

* I reserved two weeks before the mid-term and final deadline for possible schedule slips.

I divided my plan into four parts: work & improve server-offloading, work & improve R API and I tried to lay out schedules for each part in detail as much as possible.

* Last but not least, I will frequently communicate with the mentors not to deviate from the objectives. With my previous experiences combined with regular mentoring, I am confident that I will keep up with the schedule and successfully complete the project.

**MANAGEMENT OF CODING PROJECT**

**How do you propose to ensure code is submitted / tested?**

* The code will be regularly updated on Google Code repository (<https://code.google.com/p/r-interactive-graphics-via-html/>) into which I have already committed a prototype version. The final package will be submitted to CRAN for the R community to use.

**What is the communication plan with mentors?**

* I have regular access to Professor Lee and Chung Ha Sung. Furthermore, Chung Ha Sung participated in 2013 GSoC and successfully developed the first version of RIGHT. Weekly conference calls will be arranged including Dr. Jung Hoon Lee.

**Examples of similar coding problems that you have solved.**

* I took plenty of lectures in computer science, such as data structure, algorithm, computer network, operating systems, and computer architecture, etc. In these courses, I have done many software projects. Especially, the software engineering course offered an opportunity to build a software, albeit small, from design to implementation.

**TEST**

**Problem 1.** Load Theoph dataset and plot concentration (conc) versus time (Time) by subject (Subject) with at least two graphics packages in R. Points for each subject should have different colors, and a legend should be added.

**Code:**

**(a)**

rm(list = ls())

data(Theoph)

array <- unique(Theoph$Subject)

num <- length(array)

color <- colors()[sample.int(num)\*17]

plot(c(0, max(Theoph$Time)), c(0, max(Theoph$conc)), type = "p", xlab = "Time", ylab = "Conc")

for(i in 1:num) {

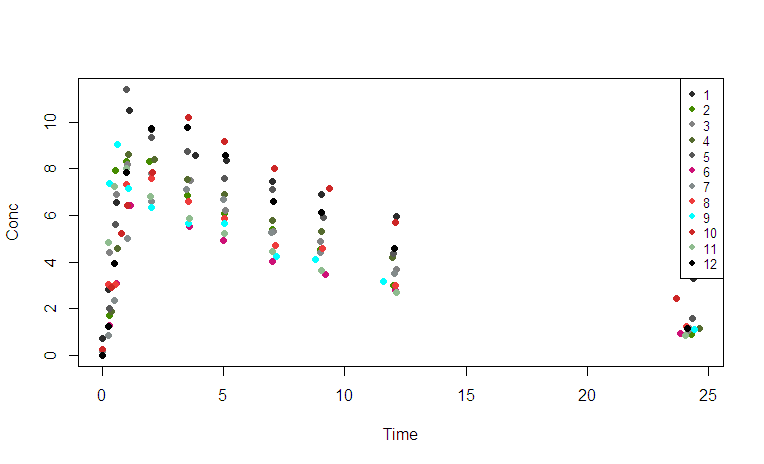
temp <- Theoph[Theoph$Subject == array[i], ]

points(temp$Time, temp$conc, pch = 19, col = color[i])

}

legend("topright", legend = array, pch = 19, col = color, cex = 0.8)

**Result :**

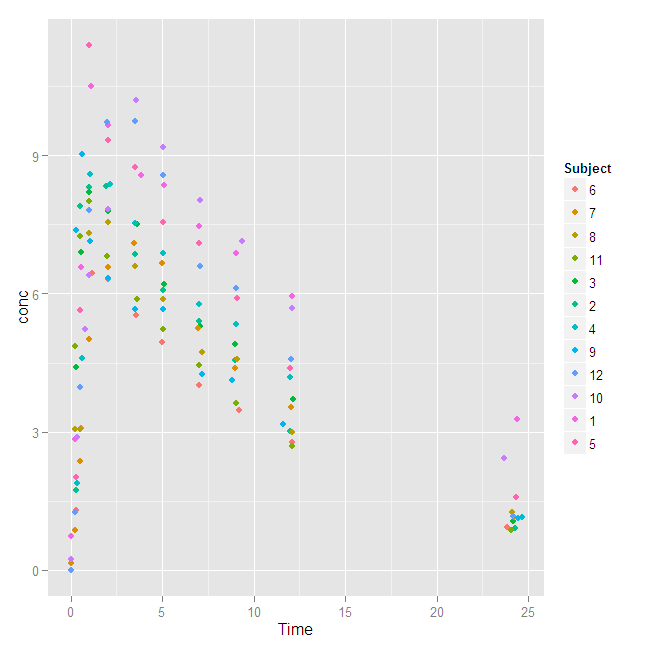


**(b)**

rm(list = ls())

library(ggplot2)

ggplot(Theoph, aes(x=Time, y=conc, color=Subject)) + geom\_point()



**Problem 2.** Re-draw the plot created in problem 1 with JavaScript using the kineticJS library (<http://kineticjs.com/>).

**Code:**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<link rel="stylesheet" href="../../JavaScript/right.css"/>

<title>RIGHT: R Interactive Graphics via HTml</title>

<script src="../../JavaScript/callback.js"></script>

<script src="../../JavaScript/kinetic-v4.7.0.js"></script>

<script src="../../JavaScript/structure.js"></script>

<script src="../../JavaScript/common.js"></script>

<script src="../../JavaScript/axis.js"></script>

<script src="../../JavaScript/color.js"></script>

<script src="../../JavaScript/dot.js"></script>

<script src="../../JavaScript/bar.js"></script>

<script src="../../JavaScript/menu.js"></script>

<script src="../../JavaScript/legend.js"></script>

</head>

<body>

<script> var mainArr1 = createMainStructure('../../JavaScript/Theoph-from-R.csv');

</script>

<div id="content" class = "right-output" >

<div id="container1" oncontextmenu="return false;"></div> </div>

<script>

var axis1 = new Axis(1, mainArr1, 'Time', 'conc', {legend:'Subject', position:'left'});

var s1 = new Dot(axis1, mainArr1, 'Time', 'conc', {});

var AllAxisObjArr = [axis1]; // this is global for redrawing in hideselected.

</script>

</body>

</html>

**Result :**

