Using R and REDCap's API to Generate Custom Project Reports

REDCap in the Rockies, 2011

Breckenridge, CO

Theresa A Scott, MS

Biostatistician III, Vanderbilt CTSA

Motivation

- Am responsible for generating one or more reports for a project.
  - Often include record level output, such as data entry checks.
  - Often include “aggregate” level output, such as tables and graphs.
  - Often *do not* include more sophisticated statistical analyses.

- Am often generating the same report(s) multiple times throughout the project’s timeline.
  - Would be great if the basic outline of the report stayed the same over time, but the “output” automatically updated itself to reflect the most current data in the project.
  - Would be great if didn’t have to manually export the data (from REDCap) each time.

Brief Introduction to R

- *Free* (ie, open source) interactive programming language and environment, created as an integrated suite of software facilities for data manipulation, simulation, calculation, and graphical display.

- Runs on Windows, MacOS, and Linux and UNIX platforms.

- Almost everything in R is done by invoking *functions*[1] that are organized into and stored in *packages*.
  - Some packages are automatically *installed* when you install R, while other "contributed" (ie, add-on) packages are available to install if you need them.
  - A package's functions are available only when the package is *loaded* in memory during an R session.

[1] Have the ability to write your own functions.

R & Literate Programming

- Literate programming: a document (eg, a project report) that is a combination of content and data analysis code.

- Instead of inserting a prefabricated graph or table into a report, the master document contains the R code necessary to generate it.
  - When run through R, all data analysis output (tables, graphs, etc.) is created on the fly and inserted into a final document.

- Allows you to create *dynamic* reports.
  - Can be easily regenerated when the data or analyses change – all of the results/tables/figures are automatically updated.
  - Facilitates *reproducible research*.

Literate Programming Options in R

- The ***Sweave*** function.
  - Part of the (base) “utils” package.
  - Allows you to embed R code in LaTeX documents.
  - Two-step process:
    - (1) Sweave the “weaved” document, which generates a .tex file.
    - (2) Compile the .tex file in LaTeX (to generate a DVI, PS, or PDF file).
  - Additionally need LaTeX (or MikTeX).

- The ***brew*** function.
  - Part of the (add-on) “brew” package.
  - Allows you to embed R code in HTML (and other text) documents.
  - Only one-step process (ie, nothing else needed); generates (HTML) file.

Review of REDCap's API

- □ API = "*A*pplication *P*rogramming *I*nterface"

- □ An interface that allows external applications to connect to REDCap remotely in order to accomplish a task.

- □ Currently possible tasks: *automated data exports* and imports from a specified REDCap project.

- □ Programs talk to the REDCap API over HTTP, the same protocol that your browser uses to visit and interact with web pages.

R through REDCap's API

- Demonstration example:
  - REDCap project created to capture (dummy) data from a 2-arm "randomized" clinical trial.

- Key points to making it work:
  - Obtained *token* that is specific to my username for my specific project from my REDCap Administrator – referenced in R code.
  - Utilized functions in several R "add-on" packages[1]:
    - "brew" for report generation.
    - "RCurl" and "bitops" for API export task.
    - "Hmisc" and "xtable" for table generation.
  - Execute code in an R session – report file automatically generated.

[1] Already installed, but loaded in report code.

Goals for the future

- Functionality that will allow a front-end REDCap user to “click a button” from within their REDCap project that would automatically execute the R code through the API and open the generated report.
  - ie, front-end REDCap user never directly interacts with R or the API.

- Functionality for users in specific Data Access groups to generate a project report on their subset of records only (ie, for multi-site clinical trials and/or statewide quality improvement collaborative).

Learning more…

- R:
  - Main website is http://www.r-project.org and can download from http://www.cran.r-project.org.
  - Both have links to a slew of manuals, books, and contributed documents for learning how to use R.
  - My “Intro to R” lectures on my website: http://biostat.mc.vanderbilt.edu/TheresaScott
  - Can also use R via “R Studio” (http://www.rstudio.org/).
    - New integrated development environment (IDE) for R.
- Brew and Sweave:
  - “Reproducible Research” link under the “Task Views” section of the http://www.cran.r-project.org webpage.
  - My “Reproducible Research With R, LaTeX, & Sweave” lecture on my wesbite (see above).

How to execute the R files

- The “Report_via_API.html” file is the HTML report file that contains both the content of the report (written in HTML) and the R code chunks that will add the additional report output.
- The “brew_api_driver.R” file contains the 3 lines of R code that you need to generate the “report.html” compiled HTML report using the “brew” function.
- NOTE: After installing R, you also need to install the “brew”, “bitops”, “Rcurl”, “Hmisc”, and “xtable” packages.
- NOTE 2: You will need to modify the “setwd” expression in the “brew_api_driver.R” file to the directory/folder where you have saved the “brew_api_driver.R” and “Report_via_API.html” files.
- From an R command prompt, submit the 3 lines of code from the “brew_api_driver.R” file and hit the “Return” key.

- If you need any help, feel free to email me: theresa.scott@vanderbilt.edu.