Flexible report generation and literate programming using R and Python's docutils module

Abhijit Dasgupta[1,*]

1. Principal Statistician, ARAASTAT, Germantown, Maryland 20876 *Contact author: adasgupta@araastat.com
Currently a consultant at NIAMS, NIH, Bethesda MD 20892.

Keywords: Literate programming, Python, OpenOffice.org, brew, Hmisc

Literate programming has been well-established in R using `Sweave`[1]. There are two major drawbacks to `Sweave` – (i) it requires one to use LaTeX for the source, and (ii) the end-product is in a non-editable format like PDF. Many of us and our collaborators typically write using Microsoft Office or similar desktop software, and it has been difficult to produce compatible literate programming output from R. Some attempts have been made to allow this using **R2HTML**[2] and **odfWeave**[3], and some good suggestions have been made on converting `Sweave`-produced documents to Word-compatible formats (see [4]). However, the results are often not satisfactory. The requirement of using LaTeX as the source code for **Sweave** has also hindered its widespread use for automatic report generation and documentation, since LaTeX has quite a steep learning curve.

reStructured Text(rSt)[5] is a text markup syntax which is easy to read, learn and format. It can be parsed using the **docutils**[6] module of Python[7] into various formats, including LaTeX, PDF, XML, HTML, and ODF (the last using the available Python script `rst2odt`[8]). reStructured text provides a very powerful, flexible and extensible platform for creating web pages and stand-alone documents, and is the preferred method for generating Python documentation. User-defined style files can be incorporated into the parsing of the *rSt* source into the final format, making it very customizable. *rSt* files converted into ODF (OpenDocument Format) can then be read by OpenOffice.org Writer (`http://www.openoffice.org`) and by Microsoft Word using either the conversion facilities of OpenOffice.org or the Sun ODF Plugin (`http://www.sun.com/software/star/odf_plugin/`). *rSt* and **docutils** have the ability to format complex tables and incorporate figures, which are the two principal needs for a literate programming platform for R. They also can incorporate fairly involved formatting into the final document as well. Source code in LaTeX or XML or HTML can be incorporated into a *rSt* document for further formatting depending on the output format (LaTeX or ODF/XML or HTML, respectively).

This work proposes to use *reStructured text* as the source platform for literate programming in R using the templating package **brew**[9] in R, and using **docutils** scripts to convert the resulting document into ODF, PDF, LaTeXor HTML formats. Tables can easily be generated using a minor modification of `print.char.matrix` in the **Hmisc**[10] library. Figures can be incorporated either as PDF or PNG depending on the final format. The resultant ODF document appears superior to the results obtained using **Sweave** and various conversion methods, producing a professional-looking document for publication and collaboration. The resultant PDF document using default templates also produces very clean tables and figures, though different from the results of PDFLaTeXand **Sweave**. The entire process can be automated using either a Makefile or a script in R. I will present examples of using this approach utilizing the `summary.formula` scripts from **Hmisc**, which produce quite complex tables.

There are three main advantages of this approach over **Sweave** and LaTeX. First of all, *reStructured text* can be quickly generated and visually formatted using a standard text editor, and doesn't require a steep learning curve to produce well-formatted documents using **docutils** scripts. The flexibility of using source code depending on the output format for further formatting or customization or inclusion of mathematics is available. Secondly, this approach allows the direct creation of native ODF files which are readable and editable by Microsoft Word, using automatically generated (and even complex) tables and figures. This allows easy transmission of the report to collaborators or publishers who regularly use Microsoft Word or related desktop tools. Thirdly, the same source file (if it does not contain specialized source code like LaTeX or XML) can be used to produce LaTeX, PDF, XML or HTML output using scripts in **docutils**, thus allowing flexibility and further potential for customization of the output.

A R package for this approach is in development.

References

1. Leisch, F (2008) Sweave User Manual.
`http://www.stat.uni-muenchen.de/~leisch/Sweave`

2. Lecoutre, E (2003). The R2HTML Package. R News, Vol 3. N. 3, Vienna, Austria.

3. Kuhn, M and Weaston, S (2009). odfWeave: Sweave processing of Open Document Format (ODF) files. R package version 0.7.10.

4. Harrell, FE (2010). Converting Documents Produced by Sweave.
`http://biostat.mc.vanderbilt.edu/wiki/Main/SweaveConvert`

5. reStructuredText: Markup Syntax and Parser Component of Docutils (2006).
`http://docutils.sourceforge.net/rst.html`

6. Docutils: Documentation Utilities: Written in Python, for General- and Special-Purpose Use.
`http://docutils.sourceforge.net`

7. The Python Languange Reference (2010).
`http://docs.python.org/reference/`

8. Kuhlman, D (2006) Odtwriter for Docutils.
`http://www.rexx.com/~dkuhlman/odtwriter.html`

9. Horner, J (2007). brew: Templating Framework for Report Generation. R package version 1.0-3.

10. Harrell, FE and others (2009) Hmisc: Harrell Miscellaneous. R package version 3.7-0.
`http://CRAN.R-project.org/package=Hmisc`