Module 1 – lesson 02

Script

In lesson 01, I mentioned the important developments of literate programming and dynamic documentation. In lesson 02 we will cover these concepts in more detail.

In Lesson 02 we will review the WEB system for literate programming; the SWEAVE tool for dynamic documentation; as well as both the knitr and rmarkdown R packages which have further enhanced dynamic documentation. We will briefly review pandoc which is a universal document converter allowing you to switch between different document formats. And I will further highlight how rmarkdown implemented within RStudio is a key “HUB” for creating documents in different formats and easily changing between these formats.

As mentioned in lesson 01, Donald Knuth introduced the concept of “literate programming” about 1992. “The main idea (of literate programming) is to treat a (software/computer) program as a piece of literature, addressed to human beings rather than to a computer.” In literate programming, the idea is that the computer code is embedded within the documentation as opposed to the documentation embedded within computer code. In literate programming the code follows the structure of the documentation.

The program that Donald Knuth used to implement his idea of “literate programming” was called WEB which he introduced earlier in 1981. WEB linked the TeX typesetting or formatting system for creating documents with the Pascal computer programming language. WEB was one of the first systems to directly link documentation creation and typesetting with computer programming together. Donald Knuth liked the name WEB as it implied that WEB was a program of ideas pieced together from simple materials.

Since WEB was introduced in 1981, many other programs implementing literate programming have emerged over time including:

* CWEB also created by Donald Knuth with Silvio Levy which was adapted for the C and C++ compute language instead of Pascal
* Axiom developed by IBM
* Noweb
* Literate
* Funnel WEB
* Molly
* Codnar
* Jupyter Notebook (formerly IPython Notebook)
* R Notebooks

As already mentioned in lesson 01, in 2002 Federich Leisch released the SWEAVE program for dynamic documentation generation. SWEAVE allows R code to be embedded within LaTeX documents. LaTeX is the more modern version of the previously mentioned TeX typesetting program used by Donald Knuth.

The really exciting feature of literate programming and dynamic documentation is highlighted here in what Friedrich Leisch stated about SWEAVE. Since the underlying computer code is wholly integrated within the document itself, anytime there are changes to the underlying data or analyses or code, the report itself is automatically updated ON THE FLY!!

It is worth noting, that the original implementation of SWEAVE uses the R statistical programming language, there is also a version of SWEAVE implemented for use with the SAS statistical software. This is called SASweave, which was released in 2007 and developed by Russell V. Lenth from the University of Iowa.

The next evolution of these ideas for literate programming and dynamic documentation have emerged from the R programming and RStudio communities.

In 2012 Yihui Xie (yeewhay she) released the R package called “knitr”. This package was inspired by SWEAVE. Similar to SWEAVE “knitr” combines R code with text typesetting for producing documents. “knitr” does work with LaTeX similar to SWEAVE, but “knitr” also works with rmarkdown which uses simple text markup syntax based on the original “markdown” package. We discussed the “markdown” package developed by John Gruber in 2004 previously in lesson 01.

“Rmarkdown” itself was fully released in 2014. The original objective of rmarkdown was creating documents for the “internet” or “web” by creating HTML formatted documents.

However, the “rmarkdown” package also leverages Pandoc for creating an even wider array of documentation formats – including:

* The DOC format such as used my Microsoft WORD or Google Docs
* The ODT format used by Libre Office
* The PDF format and others like
* EPUB for electronic-books
* Slide shows like this one using HTML5
* As well as the original TeX document formats and related TeX based slide formats like Beamer

In this course, you will not interact directly with Pandoc. However, since 2015 Pandoc has been bundled with RStudio – so when you install RStudio you get the functionality of Pandoc built in. If you would like to learn more about Pandoc itself, you can visit their website at pandoc.org. Since pandoc can convert between so many different document formats it is often called the “swiss army knife” for document conversion. As you can see, Pandoc is very versatile allowing for converting between HTML web-based formats, word processor type formats, electronic publishing or EPUB formats, presentation slide based format, publication layout formats, TeX based formats, and many others…

The RStudio Interactive Development Environment becomes our central “HUB” for combining the capabilities of:

* The great packages “knitr” and “rmarkdown”
* with the built-in functionality of Pandoc for document conversion
* plus the fantastic analysis and graphics capabilities of the R programming language.

From the RStudio interface, we can access all of this functionality and create documents on the fly in multiple formats for multiple end uses and products.

Next in lesson 03, we will cover key reproducible principles and practices and show some examples of what is possible with reproducible documents and templates.