

Using Sweave with LyX

How to lower the \LaTeX /Sweave learning curve

by Gregor Gorjanc

Introduction

\LaTeX ([\$\LaTeX\$ Project, 2005](#)) is a powerful typesetting language, but some people find that acquiring a knowledge of \LaTeX presents a steep learning curve in comparison to other “document processors.” Unfortunately this extends also to “tools” that rely on \LaTeX . Such an example is Sweave ([Leisch, 2002](#)), which combines the power of R and \LaTeX using literate programming as implemented in `noweb` ([Ramsey, 2006](#)). Literate programming is a methodology of combining program code and documentation in one (source) file. In the case of Sweave, the source file can be seen as a \LaTeX file with parts (chunks) of R code. The primary goal of Sweave is not documenting the R code, but delivering results of a data analysis. \LaTeX is used to write the text, while R code is replaced with its results during the process of compiling the document. Therefore, Sweave is in fact a literate reporting tool. Sweave is of considerable value, but its use is somewhat hindered by the steep learning curve needed to acquire \LaTeX .

The R package `odfWeave` ([Kuhn, 2006](#)) uses the same principle as Sweave, but instead of \LaTeX uses an XML-based markup language named Open Document Format (ODF). This format can be easily edited in OpenOffice. Although it seems that `odfWeave` solves problems for non- \LaTeX users, \LaTeX has qualities superior to those of OpenOffice. However, the gap is getting narrower with tools like `OOo \LaTeX` ([Piroux, 2005](#)), an OpenOffice macro for writing \LaTeX equations in OpenOffice, and `Writer 2 \LaTeX` ([Just, 2006](#)), which provides the possibility of converting OpenOffice documents to \LaTeX . \LaTeX has existed for decades and it appears it will remain in use. Anything that helps us to acquire and/or use \LaTeX is therefore welcome. LyX ([LyX Project, 2006](#)) definitely is such tool.

LyX is an open source document preparation system that works with \LaTeX and other “companion” tools. In short, I see LyX as a “Word”-like WYSIWYM (What You See Is What You Mean) front-end for editing \LaTeX files, with excellent import and export facilities. Manuals shipped with LyX and posted on the wiki site (<http://wiki.lyx.org>) give an accessible and detailed description of LyX, as well as pointers to \LaTeX documentation. I heartily recommend these resources for studying LyX and \LaTeX . Additionally, LyX runs on Unix-like systems, including MacOSX, as well as on MS Windows. The LyX installer for MS Windows provides a neat way to install all the tools that are needed to work with \LaTeX in general.

This is not a problem for GNU/Linux distributions since package management tools take care of the dependencies. \TeX Live ([\$\TeX\$ Live Project, 2006](#)) is another way to get \LaTeX and accompanying tools for Unix, MacOSX, and MS Windows. LyX is an ideal tool for those who may struggle with \LaTeX , and it would be an advantage if it could also be used for Sweave. [Johnson \(2006\)](#) was the first to embark on this initiative. I have followed his idea and extended his work using recent developments in R and LyX.

In the following paragraphs I give a short tutorial “LyX & Sweave in action”, where I also show a way to facilitate the learning of \LaTeX and consequently of Sweave. The section “LyX customisation” shows how to customise LyX to work with Sweave. I close with some discussion.

LyX and Sweave in action

In this section I give a brief tutorial on using Sweave with LyX. You might also read the “Introduction to LyX” and “The LyX Tutorial” manuals for additional information on the first steps with LyX. In order to actively follow this tutorial you have to customise LyX as described in the section “LyX customisation”.

Open LyX, create a new file with the `File -> New` menu and save it. Start typing some text. You can preview your work in a PDF via the `View -> PDF (*)` menu, where * indicates one of the tools/routes (`latex`, `pdflatex`, etc.) that are used to convert \LaTeX file to PDF. The availability of different routes of conversion, as well as some other commands, depend on the availability of converters on your computer.

The literate document class

To enable literate programming with R you need to choose a document class that supports this methodology. Follow the `Document -> Settings` menu and choose one of the document classes that indicates Sweave, say “article (Sweave noweb)”. That is all. You can continue typing some text.

Code chunk

To enter R code you have to choose an appropriate style so that LyX will recognise this as program code. R code typed with a standard style will be treated as standard text. Click on the button “Standard” (Figure 1 — top left) and choose a scrap style, which is used for program code (chunks) in literate programming documents. You will notice that now the text you type has a different colour (Figure 1). This is an indicator that you are in a paragraph with a scrap style. There are different implementations of literate

programming. Sweave uses a noweb-like implementation, where the start of a code chunk is indicated with `<<>>=`, while a line with `@` in the first column indicates the end of a code chunk (Figure 1). Try entering:

```
<<myFirstChunkInLyX>>=
xObs <- 100; xMean <- 10; xVar <- 9
x <- rnorm(n=xObs, mean=xMean, sd=sqrt(xVar))
mean(x)
@
```

Did you encounter any problems after hitting the ENTER key? LyX tries to be restrictive with spaces and new lines. A new line always starts a new paragraph with a standard style. To keep the code “together” in one paragraph of a scrap style, you have to use CTRL+ENTER to go onto a new line. You will notice a special symbol (Figure 1) at the end of the lines marking unbroken newline. Now write the above chunk of R code, save the file and preview a PDF. If the PDF is not shown, check the customisation part or read further about errors in code chunks. You can use all the code chunk options in the `<<>>=` markup part. For example `<<echo=FALSE, fig=TRUE>>=`, will have an effect of hiding output from R functions, while plots will be produced and displayed.

Inline code chunks

LyX also supports the inclusion of plain L^AT_EX code. Follow the Insert → TeX Code menu, or just type CTRL+L and you will get a so-called ERT box (Figure 1) where you can type L^AT_EX code directly. This can be used for an inline code chunk. Create a new paragraph, type some text and insert `\Sexpr{xObs}` into the ERT box. Save the file and check the result in a PDF format. This feature can also be used for `\SweaveOpts{}` directives anywhere in the document. For example, `\SweaveOpts{echo=FALSE}` will suppress output from all R functions after that line. ERT boxes are advantageous since you can start using some L^AT_EX directly, but you can still produce whole documents without knowing the rest of the L^AT_EX commands that LyX has used.

Equations

Typing mathematics is one of the greatest strengths of L^AT_EX. To start an equation in LyX follow the Insert → Math → Inline/Display Formula menu or use CTRL+M and you will get an equation box. There is also a maths panel to facilitate the typing of symbols. You can also type standard L^AT_EX commands into the equation box and, say, `\alpha` will be automatically replaced with α . You can also directly include an inline code chunk in an equation, but note that backslash in front of `Sexpr` will not be displayed as can be seen in Figure 1.

Floats

A figure float can be filled with a code chunk and Sweave will replace the code chunk “with figures”. How can we do this with LyX? Follow the Insert → Float → Figure menu and you will create a new box — a figure float. Type a caption and press the ENTER key. Choose the scrap style, insert the code chunk provided below (do not forget to use CTRL+ENTER), save the file, and preview in PDF format.

```
<<mySecondChunkInLyX, fig=TRUE>>=
hist(x)
@
```

If you want to center the figure, point the cursor at the code chunk, follow the Edit → Paragraph Setting menu and choose alignment. This will center the code and consequently also the resulting figure. Alignment works only in LyX version 1.4.4 and later. You will receive an error with LyX version 1.4.3. If you still have LyX version 1.4.3, you can bypass this problem by retaining the default (left) alignment and by inserting L^AT_EX code for centering within a float, say `\begin{center}` above and `\end{center}` below the code chunk. Check the section “LyX customisation” for a file with such an example.

Errors in code chunks

If there are any errors in code chunks, the compilation will fail. LyX will only report that an error has occurred. This is not optimal as you never know where the error occurred. There is a Python script `listerrors` shipped with LyX for this issue. Unfortunately, I do not know how to write an additional function for collecting errors from the R CMD Sweave process. I will be very pleased if anyone is willing to attempt this. In the meantime you can monitor the weaving process if you start LyX from a terminal. The weaving process will be displayed in a terminal as if R CMD Sweave is used (Figure 1, bottom right) and you can easily spot the problematic chunk.

Import/Export

You can import Sweave files into LyX via the File → Import → Sweave... menu. Export from LyX to Sweave and to other formats also works similarly. If you want to extract the R code from the document — i.e., tangle the document — just export to R/S code. Exported files can be a great source for studying L^AT_EX. However, this can be tedious, and I find that the View menu provides a handy way to examine L^AT_EX source directly. Preview of L^AT_EX and Sweave formats will work only if you set up a viewer/editor in the ‘preferences’ file (Figure 3) as shown in the following section. Do something in LyX and take a look at the produced L^AT_EX file via the View menu. This way you can easily become acquainted with L^AT_EX.

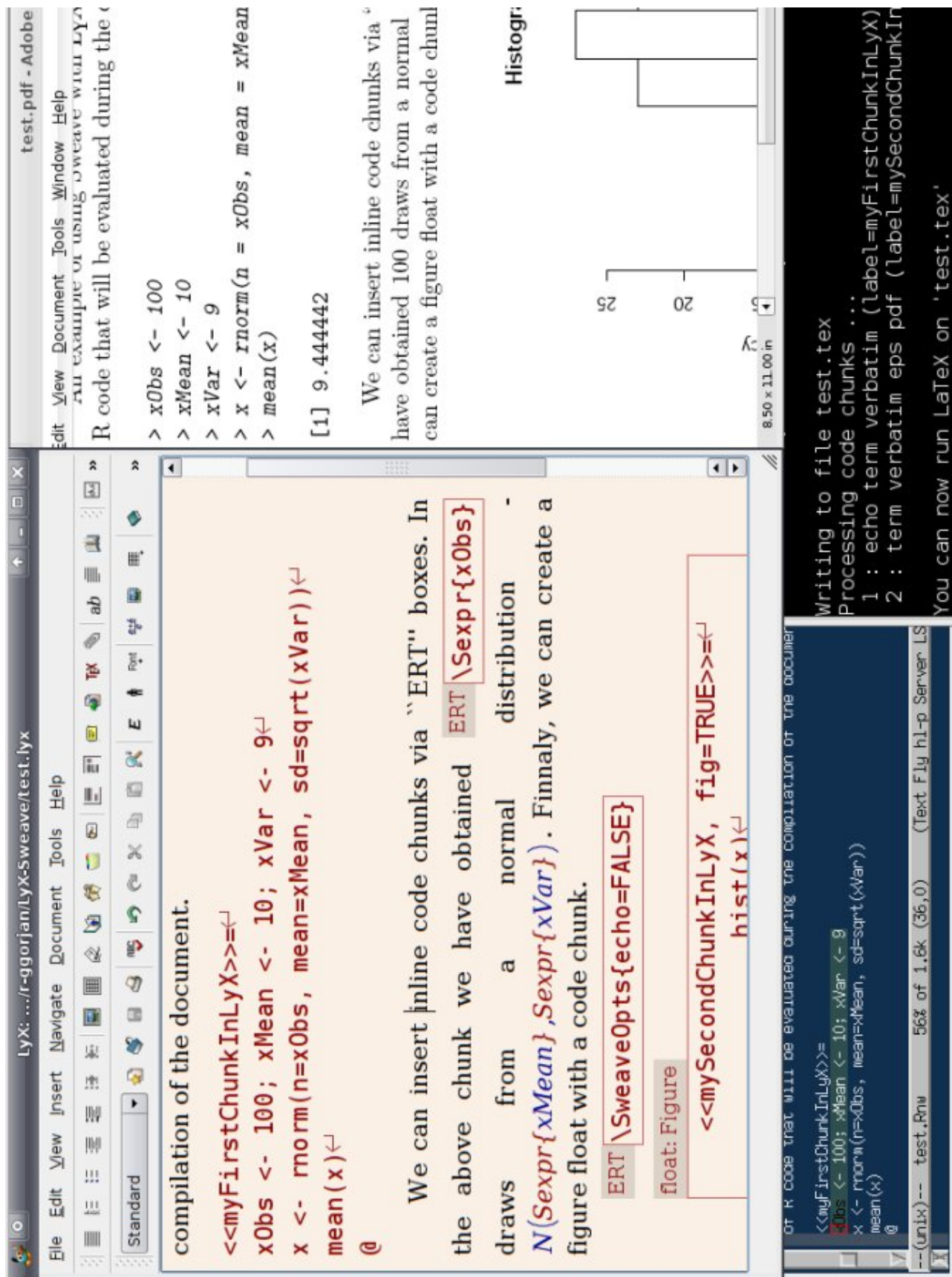


Figure 1: Screenshot of LyX with Sweave in action: LyX GUI (top-left), produced PDF (top-right), source code (Sweave) in an editor (bottom-left), and echo from weaving in a terminal (bottom-right)

In LyX version 1.5 the user can monitor L^AT_EX code instantly in a separate window. Users of LyX can therefore easily become acquainted with L^AT_EX and there should be even less reason not to use Sweave.

LyX customisation

LyX already supports noweb-like literate programming as described in the “Extended LyX Features” manual. Unfortunately, the default implementation does not work with R. To achieve this, LyX needs to be customised to use R for weaving (replacing R code with its output) and tangling (extracting program code), while LyX will take care of the conversion into the chosen output format, for example, PostScript, PDF, etc. LyX can convert to, as well as from, many formats, which is only a matter of having proper converters. For example `latex` is used to convert a L^AT_EX file to DVI format, `dvips` is used to convert a DVI file to PostScript, and you can easily deduce what the `ps2pdf` converter does. Of course, `pdflatex` can also be used to directly convert L^AT_EX to PDF. So, the idea of providing Sweave support to LyX is to specify a converter (weaver) of a Sweave file that will be used for the evaluation of R code, replacing it with the results in the generated L^AT_EX file. Additionally, a tangler needs to be specified if only the extraction of R code is required. I describe such customisation in this section, which is deliberately detailed so that anyone with interest and C++ experience could work with the LyX team on direct support of Sweave. I also discuss a possible way for this in the subsection “Future work”.

Users can customise LyX via the Tools → Preferences menu or via configuration files. Although menus may be more convenient to use, I find that handling a configuration file is easier, less cluttered and better for the customisation of LyX on different machines. Since the readers of this newsletter already know how to work with R code, the handling of another ASCII file will not present any problems. The use of menus in LyX should be obvious from the given description. Configuration files for LyX can be saved in two places: the so-called library and the user directory. As usual, the settings in the user directory take precedence over those in the library directory and I will show only the customisation for the user. The manual “Customizing LyX: Features for the Advanced User” describes all LyX customisation features as well as system-wide customisation. The configuration file in the user directory is named ‘preferences’. **Formats**, **converters**, and **document classes** need to be customised to enable Sweave support in LyX. I will describe each of these in turn. Skip to the subsection “Install” on page 7, if you are not interested in the details.

Formats

LyX formats describe general information about file formats. The default specification for the L^AT_EX file format is shown in Figure 2. This specification consists of the following fields:

- format name ("latex");
- file extension ("tex");
- format name that is displayed in the LyX GUI ("Latex (Plain)");
- keyboard shortcut ("L");
- viewer name ("");
- editor name ("");
- type of the document and vector graphics support by the document ("document").

Literate programming in LyX is implemented via the `literate` file format. The latter needs to be modified to work with R, and a new file format for R code must be introduced. The name `literate` must be used as this is a special file format name in LyX for literate programming based on the noweb implementation. The entries in the ‘preferences’ file for a modified `literate` file format and a new `r` file format are shown in Figure 3. The values in separate fields are more or less obvious — editor stands for your favourite editor such as Emacs, Kate, Notepad, Texmaker, Tinn-R, vi, WinEdt, Wordpad, etc. It is very useful to define your favourite editor for both the viewing and the editing of Sweave, R, `latex`, and `pdflatex` file formats. This provides the possibility of viewing the file in these formats from LyX with only two clicks, as noted in the “LyX & Sweave in action” section.

Converters

I have already mentioned that LyX has a powerful feature of converting between various file formats with the use of external converter tools. For our purpose, only tools to weave and tangle need to be specified, while LyX will take care of all other conversions. To have full support for Sweave in LyX the following conversions are required:

- convert (import) the Sweave file into a LyX file with R chunks;
- convert (weave) the LyX file with R chunks to a specified output format (L^AT_EX, PostScript, PDF, etc.);
- convert (tangle) the LyX file with R chunks to a file with R code only; and
- convert (export) LyX file with R chunks to a Sweave file.

```
\format "latex" "tex" "Latex (Plain)" "L" "" "" "document"
```

Figure 2: The default format specification for a \LaTeX file

```
#
# FORMATS SECTION #####
#
\format "literate" "Rnw" "Sweave"          "" "editor" "editor" "document"
\format "r"      "R"      "R/S code"      "" "editor" "editor" "document"
\format "latex"  "tex"    "LaTeX (plain)"  "" "editor" "editor" "document"
\format "pdflatex" "tex"  "LaTeX (pdflatex)" "" "editor" "editor" "document"

#
# CONVERTERS SECTION #####
#
\converter "literate" "r"          "R CMD Stangle $$i" ""
\converter "literate" "latex"     "R CMD Sweave $$i" ""
\converter "literate" "pdflatex"  "R CMD Sweave $$i" ""
```

Figure 3: Format and converter definitions for Sweave support in \LaTeX

The first task can be accomplished with \LaTeX 's import utility tool `tex2lyx` and its option `-n` to convert a literate programming file, in our case a Sweave file, to the \LaTeX file format. This can be done either in a terminal “by hand” (`tex2lyx -n file.Rnw`) or via the File \rightarrow Import menu within \LaTeX . No customisation is required for this task. `tex2lyx` converts the literate programming file to the \LaTeX file format with two minor technicalities of which it is prudent to be aware. The first one is that \LaTeX uses the term `scrap` instead of `chunk`. This is due to a historical reason and comes from another literate programming tool named `noweb` (Briggs et al., 2002). I shall use both terms (`scrap` and `chunk`) interchangeably to refer to the part of the document that contains the program code. Another technicality is related to the `\documentclass` directive in a \LaTeX /Sweave file. At the time of writing, \LaTeX provides `article`, `report` and `book` \LaTeX classes for literate programming. These are provided via document classes that will be described later on.

When converting a \LaTeX file with R chunks to other formats, the information on how to weave and possibly also tangle the file is needed. The essential part of this task is the specification of R scripts `Sweave` and `Stangle` in a ‘preferences’ file as shown in Figure 3. These scripts are part of R from version 2.4.0. Note that two converters are defined for weaving: one for `latex` and one for the `pdflatex` file format. This way both routes of \LaTeX conversion are supported — i.e., $\LaTeX \rightarrow$ PostScript \rightarrow PDF for the `latex` file format, and $\LaTeX \rightarrow$ PDF for the `pdflatex` file format. The details of weaving and tangling processes are described in the “Extended \LaTeX Features” manual.

Document classes

\LaTeX uses layouts for the definition of environments/styles, for example the standard layout/style for normal text and the `scrap` layout/style for program code in literate programming. Layout files are also used for the definition of document classes, sometimes also called text classes. Document classes with literate support for the `article`, `report` and `book` \LaTeX document classes already exist. The definitions for these files can be found in the ‘layout’ subdirectory of the \LaTeX library directory. The files are named ‘literate-article.layout’, ‘literate-report.layout’ and ‘literate-book.layout’. That is the reason for the mandatory use of the `literate` file format name as described before in the formats subsection. All files include the ‘literate-scrap.inc’ file, where the `scrap` style is defined. The syntax of these files is simple and new files for other document classes can be created easily. When \LaTeX imports a literate programming file it automatically chooses one of these document classes, based on a \LaTeX document class.

The default document classes for literate programming in \LaTeX were written with `noweb` in mind. There are two problems associated with this. The default literate document classes are available to the \LaTeX user only if the ‘noweb.sty’ file can be found by \LaTeX during the configuration of \LaTeX — done during the first start of \LaTeX or via the Tools \rightarrow Reconfigure menu within \LaTeX . This is too restrictive for Sweave users, who require the ‘Sweave.sty’ file. Another problem is that the default literate class does not allow aligning the `scrap` style. This means that the R users cannot center figures.

To avoid the aforementioned problems, I provide modified literate document class files that provide a smoother integration of Sweave and L^AT_EX. The files have the same names as their “noweb” originals.

The user can insert R code into the Sweave file with noweb- like syntax

```
<<>>=
someRCode
@
```

or L^AT_EX-like syntax

```
\begin{Scode}
someRCode
\end{Scode}
```

or even a mixture of these two (Leisch, 2002). L^AT_EX could handle both types, but L^AT_EX’s definition of the style of L^AT_EX-like syntax cannot be general enough to fulfil all the options Sweave provides. Therefore, only noweb-like syntax is supported in L^AT_EX. Nevertheless, it is possible to use L^AT_EX-like syntax, but one has to resort to the use of plain L^AT_EX markup.

L^AT_EX has been patched to incorporate the `\SweaveSyntax{}`, `\SweaveOpts{}`, `\SweaveInput{}`, `\Sexpr{}` and `\Scoderef{}` commands. These commands will be handled appropriately during the import of the Sweave file into L^AT_EX. The same holds for the L^AT_EX environment `Scode`, but the default layout in L^AT_EX used for this environment is not as useful as the noweb-like syntax.

“Install”

At least L^AT_EX version 1.4.4 and R version 2.4.0 are needed. Additionally, a variant of the Unix shell is needed. All files (‘preferences’, ‘literate-article.layout’, ‘literate-report.layout’, ‘literate-book.layout’, and ‘literate-scrap.inc’) that are mentioned in this section are available at <http://cran.r-project.org/contrib/extra/lyx>. There are also other files (‘test.lyx’, ‘Sweave-test-1.lyx’, and ‘template-vignette.lyx’) that demonstrate the functionality. Finally, the ‘INSTALL’ file summarises this subsection and provides additional information about the Unix shell and troubleshooting for MS Windows users. Follow these steps to enable use of Sweave in L^AT_EX:

- find the so-called L^AT_EX user directory via the Help → About L^AT_EX menu within L^AT_EX;
- save the ‘preferences’ file in the L^AT_EX user directory;
- save the ‘literate-*.‘ files to the ‘layouts’ subdirectory of the L^AT_EX user directory;
- assure that L^AT_EX can find and use the ‘Sweave.sty’ file (read the T_EX path system subsection if you have problems with this);

- start L^AT_EX and update the configuration via the Tools → Reconfigure menu; and
- restart L^AT_EX.

It is also possible to use L^AT_EX version 1.4.3, but there are problems with the alignment of code chunk results in floats. Use corresponding files from the ‘lyx-1.4.3’ subdirectory at <http://cran.r-project.org/contrib/extra/lyx>. Additionally, save the ‘syntax.sweave’ file in the L^AT_EX user directory.

T_EX path system

It is not the purpose of this article to describe L^AT_EX internals. However, R users who do not have experience with L^AT_EX (the intended readership) might encounter problems with the path system that L^AT_EX uses and I shall give a short description to overcome this. So far I have been referring to L^AT_EX, which is just a set of commands at a higher level than “plain” T_EX. Both of them use the same path system. When you ask T_EX to use a particular package (say Sweave with the command `\usepackage{Sweave}`), T_EX searches for necessary files in T_EX paths, also called `texmf` trees. These trees are huge collections of directories that contain various files: packages, fonts, etc. T_EX searches files in these trees in the following order:

- the root `texmf` tree such as ‘/usr/share/texmf’, ‘c:/texmf’ or ‘c:/Program Files/TEX/texmf’;
- the local `texmf` tree such as ‘/usr/share/local/texmf’, ‘c:/localtexmf’ or ‘c:/Program Files/TEX/texmf-local’; and
- the personal `texmf` tree in your home directory,

where TEX is a directory of your T_EX distribution such as MiK_TE_X (Schenk, 2006). R ships ‘Sweave.sty’ and other T_EX related files within its own `texmf` tree in the ‘pathToInstallDirectory/share/texmf’ directory. You have to add R’s `texmf` tree to the T_EX path, and there are various ways to achieve this. I believe that the easiest way is to follow these steps:

- create the ‘tex/latex/R’ sub-directory in the local `texmf` tree;
- copy the contents of the R `texmf` tree to the newly created directory;
- rebuild T_EX’s filename database with the command `texhash` (MiK_TE_X has also a menu option for this task); and
- check if T_EX can find ‘Sweave.sty’ — use the command `kpsewhich Sweave.sty` or `findtexmf Sweave.sty` in a terminal.

Users of Unix-like systems can use a link instead of a sub-directory in a local texmf tree to ensure the latest version of R's texmf tree is used. Debian GNU/Linux and its derivatives, with R installed from official Debian packages, have this setup automatically. Additional details on the T_EX path system can be found at <http://www.ctan.org/installationadvice/>. Windows users might also be interested in notes about using MiK_TE_X with R for Windows at <http://www.murdoch-sutherland.com/Rtools/miktex.html>.

Future work

The customisation described above is not a difficult task (just six steps), but it would be desirable if L_YX could support Sweave “out of the box”. L_YX has a convenient configuration feature that is conditional on availability of various third party programs and L_AT_EX files. Sweave support for editing could be configured if ‘Sweave.sty’ is found, while R would have to be available for conversions. To achieve this, only minor changes would be needed in the L_YX source. I think that the easiest way would be to add another argument, say `-ns`, to the `tex2lyx` converter that would drive the conversion of the Sweave file to L_YX as it is done for noweb files, except that the Sweave-specific layout of files would be chosen. Additionally, the format name would have to be changed from `literate` to avoid collision with `noweb`. Unfortunately, these changes require C++ skills that I do not have.

Discussion

L_YX is not the only “document processor” with the ability to export to L_AT_EX. AbiWord, KWord, and OpenOffice are viable open source alternatives, while I am aware of only one proprietary alternative, Scientific WorkPlace (SWP) (MacKichan Software, Inc., 2005). Karlsson (2006) reviewed and compared SWP with L_YX. His main conclusions were that both L_YX and SWP are adequate, but could “learn” from each other. One of the advantages of SWP is the computer algebra system MuPAD (SciFace Software, Inc., 2004) that the user gets with SWP. L_YX has some support for GNU Octave, Maxima, Mathematica and Maple, but I have not tested it. Now Sweave brings R and its packages to L_YX, so the advantage of SWP in this regard is diminishing. Additionally, L_YX and R (therefore also Sweave) run on all major platforms, whereas SWP is restricted to Windows.

Sweave by default creates PostScript and PDF files for figures. This eases the conversion to either PostScript and/or PDF of a whole document, which L_YX can easily handle. The announced support for the PNG format (Leisch, personal communication) in Sweave will add the possibility to create lighter PDF

files. Additionally, a direct conversion to HTML will be possible. This is a handy alternative to R2HTML (Lecoutre, 2003), if you already have a Sweave source.

The current default for R package-vignette files is Sweave, and since Sweave is based on L_AT_EX, some developers might find it hard to write vignettes. With L_YX this need not be the case anymore, as vignettes can also be created with L_YX. Developers just need to add vignette-specific markup, i.e., `\VignetteIndexEntry{}`, `\VignetteDepends{}`, `\VignetteKeywords{}` and `\VignettePackage{}`, to the document preamble via the Document → Settings → LaTeX Preamble menu within L_YX. A template for a vignette (with vignette specific markup already added) is provided in the file ‘template-vignette.lyx’ at <http://cran.r-project.org/contrib/extra/lyx>. A modified layout for Sweave in L_YX also defines common L_AT_EX markup often used in vignettes, for example, `\Rcode{}`, `\Robject{}`, `\Rcommand{}`, `\Rfunction{}`, `\Rfunarg{}`, `\Rpackage{}`, `\Rmethod{}`, and `\Rclass{}`.

Summary

I have shown that it is very easy to use L_YX for literate programming/reporting and that the L_AT_EX/Sweave learning curve need not be too steep.

L_YX does not support Sweave out of the box. I describe the needed customisation, which is very simple. I hope that someone with an interest will build upon the current implementation and work with the L_YX developers on the direct support of Sweave.

Acknowledgements

I would like to thank the L_YX team for developing such a great program and incorporating patches for smoother integration of L_YX and Sweave. Acknowledgements go also to Friedrich Leisch for developing Sweave in the first place, as well as for discussion and comments. Inputs by John Fox have improved the paper.

Bibliography

- P. Briggs, J. D. Ramsdell, and M. W. Mengel. Nuweb: *A Simple Literate Programming Tool*, 2002. URL <http://nuweb.sourceforge.net>. Version 1.0b1.
- P. E. Johnson. How to use L_YX with R, 2006. URL <http://wiki.lyx.org/Lyx/LyxWithRThroughSweave>.
- H. Just. *Writer 2 L_AT_EX*, 2006. URL <http://www.hj-gym.dk/~hj/writer2latex>. Version 0.4.1d.

- A. Karlsson. Scientific workplace 5.5 and LyX 1.4.2. *Journal of Statistical Software*, 17(Software Review 1):1–11, 2006. URL <http://www.jstatsoft.org/v17/s01/v17s01.pdf>.
- M. Kuhn. Sweave and the open document format – the odfWeave package. *R News*, 6(4):2–8, 2006. URL http://CRAN.R-project.org/doc/Rnews/Rnews_2006-4.pdf.
- E. Lecoutre. The R2HTML package. *R News*, 3(3):33–36, 2003. URL http://CRAN.R-project.org/doc/Rnews/Rnews_2003-3.pdf.
- F. Leisch. Dynamic generation of statistical reports using literate data analysis. In W. Haerdle and B. Roenz, editors, *Compstat 2002 - Proceedings in Computational Statistics*, pages 575–580, Heidelberg, Germany, 2002. Physika Verlag. ISBN 3-7908-1517-9.
- TeX Live Project. *A distribution of TeX and friends*, 2006. URL <http://www.tug.org/texlive/>. Version 2005-11-01.
- MacKichan Software, Inc. *Scientific Workplace*, 2005. URL <http://www.mackichan.com>. Version 5.5.
- G. Piroux. *OOoL^AT_EX*, 2005. URL <http://ooolatex.sourceforge.net>. Version 2005-10-19.
- L^AT_EX Project. *L^AT_EX - A document preparation system*, 2005. URL <http://www.latex-project.org/>. Version 2005-12-01.
- LyX Project. *LyX - The Document Processor*, 2006. URL <http://www.lyx.org>. Version 1.4.4.
- N. Ramsey. *Noweb - a simple, extensible tool for literate programming*, 2006. URL <http://www.eecs.harvard.edu/~nr/noweb>. Version 2.11b.
- C. Schenk. *MikTeX Project*, 2006. URL <http://www.miktex.org/>. Version 2.5.
- SciFace Software, Inc. *MuPad*, 2004. URL <http://www.sciface.com>. Version 3.1.

Gregor Gorjanc
University of Ljubljana
Biotechnical faculty
Slovenia
gregor.gorjanc@bfro.uni-lj.si