Using Sweave with LyX

Yihui Xie* Gregor Gorjanc Jean-Marc Lasgouttes

August 31, 2013

Sweave is a tool in the R language (`http://www.r-project.org`) for dynamic report generation using the ideas of literate programming. R code can be mixed into a LaTeX document, and will be executed in R when the document is compiled; the results are written into the TeX output.

Since LyX 2.0, Sweave can be used out of the box with any document class using the Sweave module, which means we no longer need to modify the preferences file manually or add special layouts such as literate-article.layout and literate-book.layout like we did before. We can add the Sweave module to a LyX document from Document ▷ Settings ▷ Modules. In this manual, we demonstrate some common features of Sweave and explain how the module works. A PDF version of this manual can be found at `https://github.com/downloads/yihui/lyx/sweave.pdf`.

1 System Requirements

The Sweave module depends on the executable `Rscript`, so it has to be in the PATH variable of the system. It is usually not a problem for Linux and Mac OS users, but may be confusing to Windows users; if you do not understand what is PATH under Windows, you may either use the R script add-R-path-win.R in the repository `https://github.com/yihui/lyx` to change it automatically (not recommended; you should learn PATH anyway), or go to Tools ▷ Preferences ▷ Paths ▷ PATH prefix to add the bin path of R (recommended), which is often like C:\Program Files\R\R-x.x.x\bin and you can find it in R:

```
> R.home('bin')
```

After you have done these changes, you need to reconfigure LyX by Tools ▷ Reconfigure. If `Rscript` is not present in PATH, the Sweave module will be unavailable. It is recommended to use the latest version of R, since Sweave is still being updated. Another dependency is introduced by the LaTeX style file Sweave.sty which is shipped with R under this path (it is not on CTAN):

```
> file.path(R.home('share'), 'texmf', 'tex', 'latex')
```

MikTeX users may add the texmf directory to the list of root directories in MikTeX through MikTeX settings. Further note that Sweave.sty depends on the **fancyvrb** package and *may* depend on **ae**. If LyX cannot find Sweave.sty, it will directly copy from R, so the Sweave module works even if you do not know where is this style file.

2 Sweave Examples

Before we write an Sweave document, we can set some Sweave options; the command is `\SweaveOpts{}` in Sweave and it can be inserted in a LyX document by Insert ▷ Custom Insets ▷ Sweave Options, or just

*Department of Statistics, Iowa State University. Email: xie@yihui.name

use TEX code to write it (Insert ▷ TEX Code). For example, we can set a global option `echo=FALSE` in the beginning of a document to suppress all R code (by default, both R code and its output will show up in the LATEX document).

To write R code in LYX, you can first switch to the `Chunk` environment in the layout styles list (droplist in the top-left corner in the toolbar), or you can simply open a TEX code environment and write Sweave code chunks in it. The latter approach is recommended since it is more stable.

Some R options can be useful to the Sweave output as well, e.g. `width` and `useFancyQuotes`. The former option can adjust the width of the output, and the latter can avoid a common problem with quotes. Users are encouraged to read the Sweave manual in R for more details.

```
> ## the option 'digits' can set the (approximate) number of digits
> options(width = 70, useFancyQuotes = FALSE, digits = 4)

> ## view the Sweave vignette (requires R >= 2.13.0)
> vignette("Sweave", package = "utils")
```

A simple example that will run in any S engine:

```
> ## the integers from 1 to 10 are
> 1:10

 [1]  1  2  3  4  5  6  7  8  9 10

> ## intentionally hide the output by setting results=hide
> print(1:20)
```

We can also emulate a simple calculator:

```
> 1 + pi

[1] 4.142

> sin(pi/6)

[1] 0.5
```

Now we look at some Gaussian data:

```
> set.seed(123)  # for reproducibility
> x <- rnorm(20)
> print(x)

 [1] -0.56048 -0.23018  1.55871  0.07051  0.12929  1.71506  0.46092
 [8] -1.26506 -0.68685 -0.44566  1.22408  0.35981  0.40077  0.11068
[15] -0.55584  1.78691  0.49785 -1.96662  0.70136 -0.47279

> t1 <- t.test(x)
```

Note that we can easily integrate some numbers into standard text; this is done with the command `\Sexpr{}`, and the corresponding LYX menu is Insert ▷ Custom Insets ▷ S/R expression, or alternatively just use TEX code. For example, the third element of the vector `x` is 1.55870831414912 (i.e. `x[3]`) and the p-value of the test is 0.52. You can round these numbers using functions like *round()* in R.

Now we look at a summary of the `iris` dataset:

```
> summary(iris[,-5])
```

Table 1: Quantiles of the first four variables in the `iris` data.

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
0%	4.30	2.00	1.00	0.10
25%	5.10	2.80	1.60	0.30
50%	5.80	3.00	4.35	1.30
75%	6.40	3.30	5.10	1.80
100%	7.90	4.40	6.90	2.50

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.30   Min.   :2.00   Min.   :1.00   Min.   :0.1
 1st Qu.:5.10   1st Qu.:2.80   1st Qu.:1.60   1st Qu.:0.3
 Median :5.80   Median :3.00   Median :4.35   Median :1.3
 Mean   :5.84   Mean   :3.06   Mean   :3.76   Mean   :1.2
 3rd Qu.:6.40   3rd Qu.:3.30   3rd Qu.:5.10   3rd Qu.:1.8
 Max.   :7.90   Max.   :4.40   Max.   :6.90   Max.   :2.5
```

Usually people do not like such "raw" output, and we can polish these messy numbers with the **xtable** package. Table 1 is an example demonstrating how to use the **xtable** package with Sweave; note the chunk option `results=tex` makes sure that the R output is written out as raw LaTeX code instead of being wrapped in a special environment (`Soutput`).

Figure 1 and 2 are two simple examples of producing graphics output with Sweave.

We can also export all the R code in an Sweave document as a single R script (File ▷ Export ▷ R/S code), which is done via `R CMD Stangle`.

3 The Sweave Module

The Sweave module declares that a LyX document contains literate programming components with R, so that LyX will handle such documents according to the specific converters (see the Customization manual). The Sweave module uses the sweave converter, which essentially consists of an R script lyxsweave.R under the scripts directory of LyX. This script takes an Rnw document from LyX and call Sweave to process it to a TeX document, then LyX takes care of the rest of work.

The converter from the sweave format to LaTeX is a call to R via the command line as:

```
Rscript --no-save --no-restore $$s/scripts/lyxsweave.R $$p$$i $$p$$o $$e $$r
```

The LyX library directory `$$s` can be found from the menu Help ▷ About LyX. All the rest of options passed to R are explained in the R script, and the LyX Customization manual also has detailed explanations about `$$i`, `$$o`, `$$e` and `$$r`. General users do not need to know clearly what they mean, but here we need to explain a few issues in the R code:

1. the encoding string of the LyX document is passed to R as an command line option; possible values are `ISO-8859-15` and `UTF-8`, etc. The encoding is used for R to read the Rnw document correctly.

2. the R code chunks in the Sweave document are executed under the directory of the LyX document (if you are not sure, put `getwd()` in a code chunk to see what is the current working directory). In this case, you can put data files under the same directory and use relative paths in R to guarantee reproducibility, i.e. we do not need to write hard-coded paths which may not exist in other systems.

3. if LaTeX cannot find Sweave.sty, this R script will copy it to the temporary directory automatically where the LaTeX compilation takes place.

```
> pairs(iris, col = iris$Species)
```

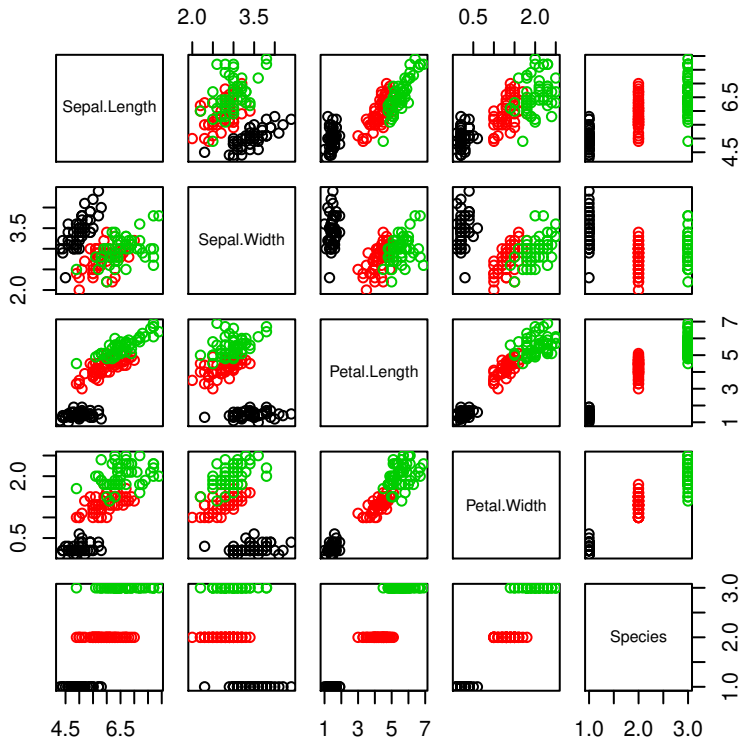

Figure 1: Scatter plot matrix of the `iris` data.

```
> par(mar = c(4, 5, 0.1, 0.3), las = 1)
> boxplot(Sepal.Length ~ Species, data = iris, horizontal = TRUE,
+     xlab = "Sepal.Length")
```

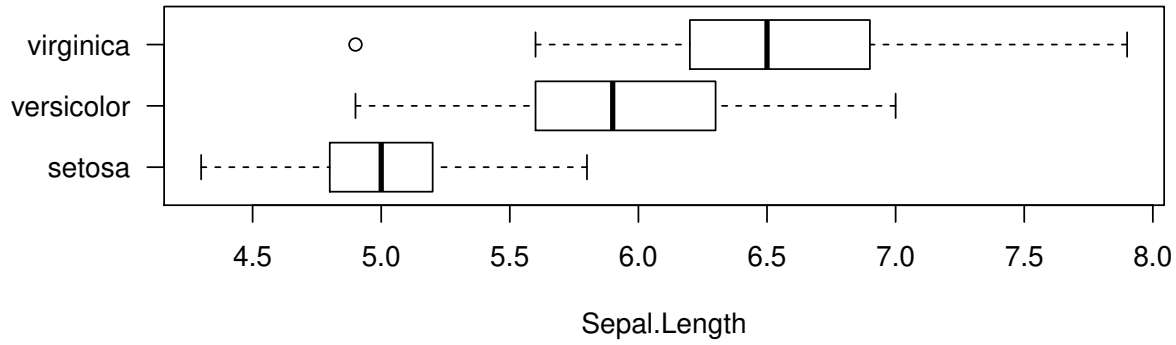

Figure 2: Boxplots of sepal length grouped by species.

4. Sweave will use the filename of the Rnw file as the prefix string for the graphics output by default; one known issue (#7551) is that if the filename (without extension) contains dots, LaTeX can fail to deal with such files in `\includegraphics{}`. Therefore, we have set the default value of the prefix to be the filename with all the dots being replaced by "-". We can also solve this problem in Sweave directly: set the Sweave option `prefix.string` to be a character string without dots, and this option will override the default value.

We have pre-specified an option `noae` for the Sweave package. This option stops Sweave from loading these two packages:

```
\RequirePackage[T1]{fontenc}
\RequirePackage{ae}
```

LyX can deal with the font encoding by default, so there is no need to ask Sweave to do this (furthermore, this may bring clashes of package options); besides, the users can load the **ae** package by themselves in the preamble if needed.

4 Trouble Shooting

Since LyX 2.0, we can see the detailed information during compilation via View ▷ View Messages. This is extremely helpful for trouble shooting – the process of R will be shown in the message panel, and we will be able to know which chunk goes wrong in cases of errors. For example, when you compile this document, you can see messages like below:

```
17:58:47.868: Processing code chunks with options ...
17:58:47.885:  1 : echo keep.source term verbatim (label=setup)
17:58:47.889:  2 : echo keep.source (label=sweave-manual)
17:58:47.908:  3 : echo keep.source term verbatim (label=print-integers)
....
17:58:47.941: Loading required package: xtable
17:58:47.976:  9 : echo term verbatim pdf (label=iris-pairs)
....
17:58:48.116: You can now run (pdf)latex on 'sweave.tex'
```

The string after `label=` is the chunk label. If any chunk causes an error in R, you will see the error message here immediately.

Another known issue is the Sweave code chunk may fail when we change the alignment of the chunk paragraph in LyX (using Paragraph settings), because LyX may add a macro like `\centering{}` before the code chunk without adding a new line, which will lead to a code chunk like this:

```
\centering{}<<xtable-demo,results=tex,echo=FALSE>>=
```

This is not a legitimate Sweave code chunk, since `<<>>=` must start in a new line with nothing before it. Therefore we recommend you to double check the real source code via View ▷ View Source when changing the alignment of a chunk paragraph, and make sure `<<>>=` appears in a new line. This explains why we manually added an empty line in the code chunk in Table 1.

5 Resources

The homepage of Sweave is at `http://www.stat.uni-muenchen.de/~leisch/Sweave/`. The CRAN task view on "Reproducible Research" has a list of Sweave-related packages which can be potentially useful; see `http://cran.r-project.org/web/views/ReproducibleResearch.html`. Questions about LyX and Sweave can be posted to the LyX mailing list (`http://www.lyx.org/MailingLists`). More LyX examples are provided at `https://github.com/yihui/lyx`.

6 FAQ

6.1 How to use Sweave with beamer?

You will end up with LaTeX errors if you use Sweave in beamer slides in LyX; the reason is the beamer frame that contains verbatim R code and output must have the `fragile` option. However, currently there is no straightforward way to do this. One way is to write raw TeX code, and the other is to use the beamer-fragile.module discussed in #7273 (put this file under the layouts directory of LyX user directory and reconfigure LyX). After including this module in your document, you should be able to use Insert ▷ Custom Insets ▷ FragileFrame, which is essentially `\begin{frame}[fragile]`.

6.2 Can I define my own R script to compile the Rnw document?

Yes, of course. First you need to understand how the R script `$$s/scripts/lyxsweave.R` works, which was introduced previously, and we recommend you to read the R source code as well (on how R takes the paths and uses *Sweave()* to handle the Rnw document). Then you put your customized lyxsweave.R under the scripts directory of your *user directory* (again, see Help ▷ About LyX). Note LyX will check the *user directory* before going to its own *library directory*; if an R script lyxsweave.R is found in the user directory, it will be used. It is not recommended to hack the script under the LyX library directory directly, since updating LyX will override you modified script (the user directory will not be affected).

Beside the support for the official Sweave in R, the **knitr** package is also supported, which is an alternative tool to Sweave; the corresponding module is named Rnw (knitr).