Using Sweave with WinEdt 5.5 or later

Duncan Murdoch

October 27, 2013

1 Setup

These instructions were originally written for version 5.5 of WinEdt. Since I wrote them, the R-Sweave macro package has appeared. I have not used it, but I believe it replaces most of what these instructios do, in WinEdt 5.5, 5.6, 6, 7 or 8. You should download it from `http://www.winedt.org/Config/modes/R-Sweave.php` and follow the instructions in the readme.txt file to install.

If you are using WinEdt 5.5 (but not newer versions), you can instead follow the instructions in Section 1 to work with WinEdt without using R-Sweave. The instructions in section 2 are about Sweave, and should work whichever way you choose to go.

1.1 Telling WinEdt to handle Sweave documents

Sweave documents are typically named *.Rnw. You want WinEdt to recognize these documents and treat them as if they are LaTeX.

First, you need to tell Windows to associate them with WinEdt:

In Explorer, right click on a *.Rnw file, and choose Open with...|Choose program...

Check the "always use the selected program" checkbox, and pick WinEdt (possibly using Browse).

Now, WinEdt will start whenever you double click on a Rnw file. You still need to tell it that this is really LaTeX.

Choose Options | Preferences | Modes and add ;*.Rnw to the end of the list of extensions for TeX files.

This will work for new .Rnw files but the current file will not be recognized. To fix it, click on the block at the bottom of the screen that lists the current mode (probable "DATA"), and change it to TeX.

1.2 Telling WinEdt to go straight to the previewer

I don't like to have to click the LaTeX button, then click the previewer button. I want the LaTeX button to take me straight to the previewer. To do this, use

Options | Execution modes

then highlight LaTeX, and check "Start viewer" and "Forward search". Do the same for PDF LaTeX. On some systems this dialog is in a different location, i.e.

Options | Configuration Wizard | Diagnosis | Execution modes

but I believe the same procedure will work once you find the dialog.

1.3 Telling WinEdt to use Sweave to process Sweave documents.

Sweave needs to run before LaTeX runs. Some people do this with Makefiles, but I don't want to get into those. The simplest way to do this is to install a small package into R.

1.3.1 Installing the patchDVI package

This package is available online at `http://r-forge.r-project.org/projects/sweavesearch`. Start R, and run the command

```
install.packages("patchDVI", repos="http://r-forge.r-project.org")
```

(In current versions of R you can use the menus to select R-forge as your repository, and install completely from the menus.)

If you don't have admin authority on your PC, you'll need to install to a local library. For example, I installed to `N:/Rlibs27` in the undergrad lab.

1.3.2 Telling WinEdt to use patchDVI

I recommend changing the Texify and pdfTexify commands, and leaving the others alone.

In

Options | Execution modes

choose Texify, and click on Browse for Executable. Find the Rterm.exe executable (version 2.7.0 or higher), and choose that.

In the Switches line, put

```
--slave -e
```

and in the Parameters line, put

```
"library(patchDVI);SweaveMiktex('%n%t', '%N.tex')"
```

The quotes are necessary!!!

Do similarly for the PDF Texify command, but use

```
"library(patchDVI);SweavePDFMiktex('%n%t', '%N.tex')"
```

as the Parameters.

A few things can go wrong in this step:

- If you don't have administrative rights on your computer, you'll need to install patchDVI into a local library, e.g. `N:/Rlibs27`. Then you might need to change `library(patchDVI)` to `library(patchDVI, lib.loc="N:/Rlibs27")`.
- You might find that your version of pdflatex doesn't support some of the options used by SweavePDFMiktex. In that case, you can probably use the command

  ```
  SweavePDFMiktex('%n%t', '%N.tex', options='')
  ```

  in place of what I used above.

1.4 Telling the previewer to jump back to WinEdt

It is handy to be able to click on a line in the previewer, and go to that line in the source document. In the past this was not possible in PDF previewers on Windows, but you can do it with the DVI previewer Yap that comes with MikTeX.

Start the previewer (e.g. by texifying something), and go to View | Options | Inverse DVI search. You should see "WinEdt (auto-detected)" as an option; if so, select it. If not, create a new entry for WinEdt, and for the command line, put in

```
"path\to\winedt.exe" "[Open(|%f|);SelPar(%l,8)]"
```

after editing the path as necessary.

Very recently the PDF previewer SumatraPDF has started to support SyncTex data to allow reverse search. This is still a little rough around the edges, but it's good news for patchDVI. To make use of this, you need to work on the bleeding edge: use the latest patchDVI version 1.4 or later, version 0.9 or better of SumatraPDF, and the latest version of MikTeX. I believe there are a few other PDF viewers that support SyncTex data, but I've never used them.

1.5 Jumping to the .Rnw file, not the .tex file

Yap uses information encoded in the DVI file to know where to jump. LaTeX puts information about the .tex file that comes out of Sweave, not the .Rnw file that went in. The patchDVI package can convert the information.

To do so, put the lines

```
\usepackage{Sweave}
\SweaveOpts{concordance=TRUE}
```

early in your .Rnw file.

2 Embedding R code in your document

2.1 Basics of Sweave

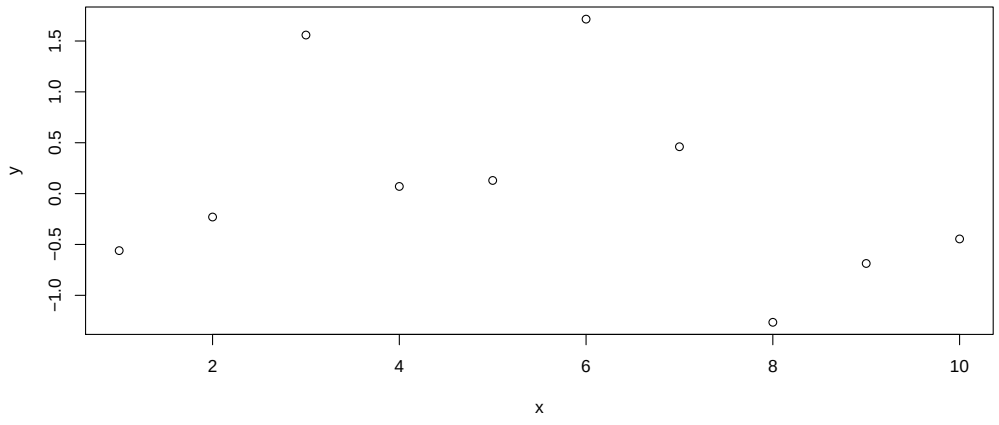
The Sweave manual gives a much more complete description; I'll just give a very short one here.

The idea of Sweave is that your document corresponds to a session of R. You embed text like

```
<<fig=TRUE>>=

set.seed(123)
x <- 1:10
y <- rnorm(10)
y
plot(x, y)

@
```

into your .Rnw file, and Sweave converts it into this:

```
> set.seed(123)
> x <- 1:10
> y <- rnorm(10)
> y

 [1] -0.56047565 -0.23017749  1.55870831  0.07050839
 [5]  0.12928774  1.71506499  0.46091621 -1.26506123
 [9] -0.68685285 -0.44566197

> plot(x, y)
```



You don't need to worry about the details of importing the code, the results or the figure; Sweave does that for you, by producing the figure in both EPS and PDF format, and generating LaTeX code like this:

```
\begin{Schunk}
\begin{Sinput}
> set.seed(123)
> x <- 1:10
> y <- rnorm(10)
> y
\end{Sinput}
\begin{Soutput}
 [1] -0.56047565 -0.23017749  1.55870831  0.07050839
 [5]  0.12928774  1.71506499  0.46091621 -1.26506123
 [9] -0.68685285 -0.44566197
\end{Soutput}
\begin{Sinput}
> plot(x, y)
\end{Sinput}
\end{Schunk}
\includegraphics{figs/-002}
```

The Schunk, Sinput and Soutput macros can be customized if you want fancy colours or other things in your document.

2.2 Recommended Sweave options

We've already seen the line `\SweaveOpts{concordance=TRUE}` above. There are other Sweave options I often use:

height=5, width=10 These allow you to specify the height and width of plots produced by R. I tend to choose values slightly larger than I want the plot to appear, because I like a slight reduction in the font sizes that results when LaTeX shrinks it.

keep.source=TRUE If you don't choose this, then R will reformat all of your carefully written source code.

prefix.string=figs/ This says to store all the figures (and the concordance) in a subdirectory called figs, so they don't mess up your main directory. You need to create that subdirectory or this will fail.

These can all be combined into one `\SweaveOpts{}` call, by separating them with commas.

I also set some options to the graphicx package, so my figures look consistent:

```
\setkeys{Gin}{width=\textwidth}
```

This says that figures should be resized so that they fit the full width of the text.

And finally, an option to R:

```
options(width=60)
```

will shorten the output lines that R produces, so they look nicer in a document. To get this to execute without showing up in your document, include it as

```
<<echo=FALSE>>=

options(width=60)

@
```

early in your document.