Using Sweave with XeLaTeX

Markus Gesmann

October 24, 2011

Contents

1 Introduction

Using R with LaTeX via Sweave [2] is a great way to create reproducible output. However, using specific fonts, e.g. your corporate fonts, can be painful with `pdflatex`. Over the last few weeks I have fallen in love with the TeX format XeLaTeX [1] and its XeTeX engine[1].

With XeLaTeX I had to overcome some hurdles, which I would like to share here:

- attaching files,
- trimming and clipping images,
- learning how to use the R package tikzDevice.

2 Moving from `pdflatex` to `xelatex`

XeLaTeX comes with all major TeX distributions such as TeX Live 2011, MacTeX

[1] A good summary of the various TeX engines is given in the Guide to LuaTeX.

2011 and MikTeX 2.9. It is still in development, therefore I strongly suggest to update your TEX distribution and packages before you start experimenting and using XELATEX.

XETEX natively supports Unicode and the input file is assumed to be in UTF-8 encoding by default. As an example, it means I can type £ instead of having to use `\pound` to get the same result. Further XETEX can use any fonts installed in the operating system without configuring TEX font metrics.

Being able to use your fonts is great, but then you also want to use the same fonts in your R-plots. Therefore we have to talk about the `tikzDevice` package [5] as well. The TikZ device enables LATEX-ready output from R graphics functions. The TikZ device creates a tex-file using the TikZ graphics language. All text in a graphic output with the tikz() function will therefore match the current font used in your document.

2.1 Using fonts

Using fonts could not be easier in XELATEX: Insert the following four lines into the preamble of your tex-file and the most commonly required packages by XELATEX will be loaded and the main font will be changed to Optima and the section font to Calibri[2].

```
\usepackage{xltxtra,fontspec,xunicode}
\usepackage{sectsty} %% change fonts for sections
\setmainfont{Optima}
\sectionfont{\fontspec{Calibri}}
```

Changing the font within your text temporarily is equally easy, as the following demonstrate:

```
\fontspec{Zapfino} %% Change to Zapfino
Hello World in Zapfino
\fontspec{Optima} %% Change back to Optima
```

Dario Taraborelli has some beautiful examples on his web page: `http://nitens.org/taraborelli/latex`.

2.2 Attaching files

To attach a file to a PDF document using LATEX with `pdflatex` I used to use the `attachfile` package. With XELATEX we have to use the package `attachfile2` [3] instead. The functionality appears to be the same, so to attach a file we write:

[2] Please note that these fonts might not be available on your system.

```
\attachfile[icon=Paperclip]{waves.pdf}{waves.pdf}
```

This results in: waves.pdf. Please note that not all PDF-viewer provide access to the attachment, if in doubt use Adobe Reader.

We can even attach the Sweave source file of this document as well: .

2.3 Trimming and clipping

Clipping and trimming of pictures in XeLaTeX requires a little more work and an up-to-date TeX-system. Let's look at the following example in Figure 1. Using

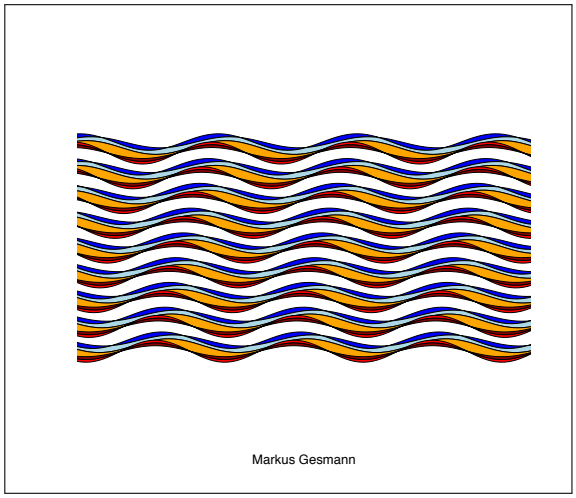

Figure 1: Original picture

`pdflatex` we can clip and trim the excess white space of the picture like this:

```
\includegraphics[trim=20mm 37mm 10mm 35mm, clip]{waves.pdf}
```

To achieve the same result with XeLaTeX and the `xetex` engine we have to use the `adjustbox` package by Martin Scharrer [4].

The following lines result in the desired output of Figure 2.

```
\begin{figure}[ht]
\begin{center}
%% cut of 35mm from the top, 37mm from the bottom,
%% 10m from the right and 21mm from the left
\adjustbox{trim=20mm 37mm 10mm 35mm, clip, width=0.6\textwidth}{
\includegraphics{waves.pdf}
}
\caption{Clipped and trimmed picture}
```

```
\end{center}
\end{figure}
```

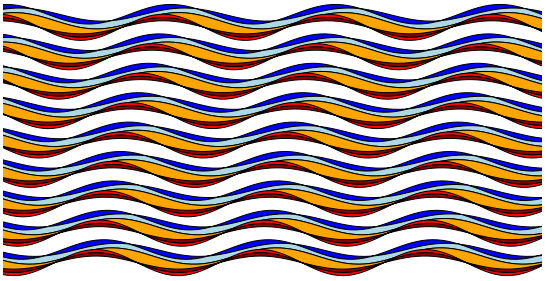

Figure 2: Clipped and trimmed picture

2.4 Using your fonts in R plots

To achieve a consistent look in our document we also need to ensure that the labels of our plots are set with the same font as the document itself.

As mentioned above, the `tikzDevice` package by Charlie Sharpsteen and Cameron Bracken provides a graphic device for R which generates tex output using the TikZ language. As a result we embed the chart not with the `includegraphics` command, but with the `input` statement. Further we have to add the `Sweave` and the `tikz` package to the preamble of our Rnw-file:

```
\usepackage{Sweave}
\usepackage{tikz}
```

As an example we want to replicate the bar plot of the blog entry *R related books: Traditional vs online publishing*.

```
> bibfile <- readLines("http://www.r-project.org/doc/bib/R-books.bib")
> pub.start.pos <- regexpr("publisher =", bibfile, perl=TRUE)
> pub.lines <- which( pub.start.pos > 0 )
> pub.split <- strsplit(bibfile[pub.lines], "[ =,]", perl=TRUE)
> publishers <- sapply(pub.split, function(x) paste(x[-c(1:5)]))
> publishers <- gsub("[{}\"),\\]", "", publishers)
> publishers <- gsub("c\\(", "", publishers)
> s=c("Springer", "Wiley", "Sage", "Chapman & Hall", "CRC press", "Servicio")
> r=c("Springer", "Wiley", "Sage", "Chapman \\& Hall/CRC",
+     "Chapman \\& Hall/CRC","Universidad de La Rioja")
> for(i in seq(along=s)){
+   publishers[regexpr(s[i], publishers, ignore.case=TRUE) > 0] <- r[i]
+ }
> RBooks <- table(publishers)
> RBooks <- RBooks[order(RBooks)]
```

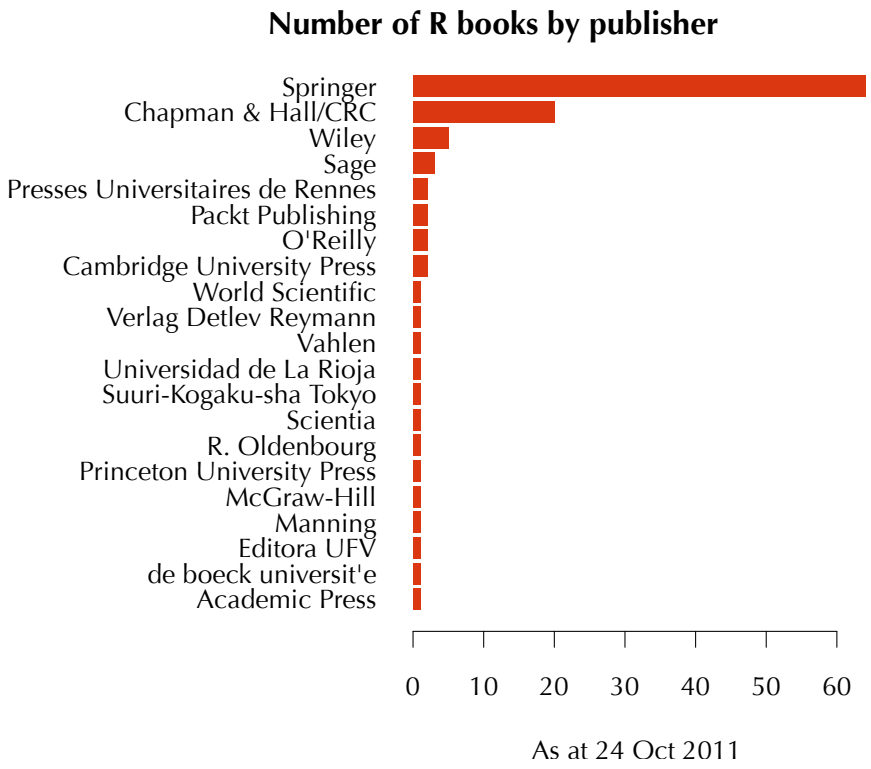

Figure 3: A barplot using the Optima font.

Creating the TikZ output works in the same way, as with PDF, PNG or any other graphic device: we create a file connection, execute the R plotting commands and close the device at the end.

```
> require(tikzDevice)
> tikz(file="myBarplot.tex", width=5.2,height=4)
> opar <- par(mar = c(4, 13, 0, 4), oma=c(0,0,2,0))
> barplot(RBooks, horiz=TRUE, las=1, col="#DC3912",
+   xlab=format(Sys.time(), "As at %d %b %Y"),
+   border=NA)
> title("Number of R books by publisher", outer=TRUE)
> par(opar)
> dev.off()
```

To insert the plot we use the figure environment and use the `input` command:

```
\begin{figure}
\input{myBarplot.tex}
\caption{A barplot using the Optima font.}
\label{barplot}
\end{figure}
```

The chart is included as Figure 3 using the Optima font, as the rest of the body text.

3 Conclusion

Using XeLaTeX requires a few minor changes in you work flow, but in combination with the `tikzDevice` package and Sweave it allows you to change the fonts in your document easily.

References

[1] Jonathan Kew. XeTeX,. Version 0.9997.5 `http://scripts.sil.org/xetex`, 2011.

[2] Friedrich Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. In Wolfgang Härdle and Bernd Rönz, editors, *Compstat 2002 --- Proceedings in Computational Statistics*, pages 575--580. Physica Verlag, Heidelberg, 2002. ISBN 3-7908-1517-9.

[3] Heiko Oberdiek. Package attachfile2: Attach files into pdf. Version 2.6. `http://www.ctan.org/pkg/attachfile2`, 2011.

[4] Martin Scharrer. Package adjustbox: Apply graphics package macros to general boxes. Version 0.6. `http://www.ctan.org/pkg/adjustbox`, 2011.

[5] Charlie Sharpsteen and Cameron Bracken. *tikzDevice: A Device for R Graphics Output in PGF/TikZ Format*, 2011. R package version 0.6.1.