The PracTEX Journal, 2008, No. 1
Article revision 2008/4/4

Learning to Sweave in APA Style

Ista Zahn

Email izahn@psych.rochester.edu
Address University of Rochester
Clinical and Social Sciences in Psychology
Meliora Hall
Rochester, NY 14627

Abstract Until recently I used Micorsoft Word and clones such as OpenOffice to write academic manuscripts, as do most in my field. The standard software toolkit for many psychology professors and graduate students also includes SPSS for performing statistical analyses, and perhaps EndNote or similar reference manager software for generating bibliographies. These tools work, but my experience suggests that LATEX based solutions have significant advantages. This article describes how to use Sweave to write LATEX documents in APA style, complete with results, tables, and figures generated by R.

1 Why I made the switch to LATEX and R

I am the only person in my department who uses LATEX and R. Because Sweave[1] simply provides a way to integrate these two programs, it follows that I am the only Sweave user as well. Why have I taken the time and effort to learn these programs instead of following the crowd and sticking with Word and SPSS? Quite simply, I made the switch because using LATEX and R is actually *easier*. It took me some time to become familiar with these programs, but after using them for a couple of months I am firmly convinced that I am more productive with these programs than I ever was with Word and SPSS.

1.1 Advantages to using LATEX and R

There are a number of major advantages to using LATEX and R rather than Word and SPSS. For example, LATEX users who need to write documents in APA style can use

[1] R is an open-source implementation of the S statistical programming language. Sweave stands for 'S-weave', i.e., weaving S code and results into a LATEX document.

`apa.cls`[6] to completely specify the final appearance of the document. Unlike the writer using Word, the LaTeX user does not need to manually set page numbers, figure out where to place the running head, affiliation information etc. The LaTeX user does not even need to fiddle with the style of the headings. This is one especially irritating aspect of APA style; because the headings used differ depending on the number of headings, one must alter headings throughout the document if the number of heading levels is changed. With LaTeX these details are handled by the computer, allowing the author to (almost) forget about formatting and concentrate on the content of the article. There are additional advantages to using LaTeX was well, such as the ability to easily create indexed `.pdf` files that can be navigated using free `PDF` viewers like Acrobat Reader, and the ability to easily create reference lists without manual formatting.

When used together with R, the LaTeX document becomes an almost self-contained environment in which the author can perform statistical analyses, present the results in standard APA style, compose the prose of the article, and create and display tables and figures. All of these activities can take place within a single computer program (your favorite text editor), reducing distractions and facilitating complete immersion in the writing process. This article documents some of my experiences using LaTeX and R, and highlights some of the features I find most useful, including automatic formatting of title pages and section headers, the creation of tables and figures, cross referencing, and the generation of APA style reference lists.

LaTeX and R are both wonderful tools in their own right. LaTeX is a powerful, flexible, and free alternative to the more standard what you-see-is-what-you-get approach to document creation, such as taken by Microsoft Word and OpenOffice. R is a powerful, extremely flexible, and free tool for performing statistical analyses and computations. It can replace costly alternatives such as SPSS or STATA, and can easily handle the relatively simple analyses I need to perform, including analysis of variance, multiple regression, and structural equation modeling. Sweave is a system that brings these two wonderful tools together. A Sweave document is basically a LaTeX document interspersed with R code.

1.2 Disadvantages to using LaTeX and R

Unfortunately, there are also drawbacks to using LaTeX. The most serious drawback I have encountered is that it can be difficult to collaborate with colleagues who do not use LaTeX. Those unfamiliar with LaTeX are unlikely to appreciate being sent a LaTeX file, and it can be difficult to convert LaTeX files to a format Word can read. If your collaborators (or perhaps more importantly, journal editors) won't accept LaTeX documents, you are going to have to figure out a way to do this conversion. I provide instructions for how to do this in Section 6.

The main disadvantage to using R is the startup cost in terms of learning the syntax. There are graphical user interfaces available (such as R Commander[2]), but generally speaking, if you want to use R you are going to have to learn to write R commands. In my case it took about two months to become comfortable enough with R to let my SPSS license lapse.

2 Formatting your document in APA style

Because I need to write documents in APA style, I am extremely fond of the APA document class[6]. Although this article is primarily about Sweave, I'm going to take a moment to highlight some of the features of this of `apa.cls` that I find most useful.

To use `apa.cls`, write \documentclass[man]{apa} in your preamble. This will cause your document to be typeset in APA manuscript style. If you want to see approximately what the published article will look like, you can use the *journal* mode by replacing [man] with [jou].

2.1 Multiple authors and affiliations

`apa.cls` supports multiple authors by defining the commands author, twoauthors, and so on, up to sixauthors. For example, if you have three authors, specify the author information as follows:

```
\threeauthors{First Author}{Second Author}{Third Author}
```

If you specify multiple authors, you must specify an affiliation for each of them. For the example above, you could specify

```
\threeaffiliations{Institution A}{Institution B}{Institudion C}
```

You can specify up to six affiliations.

Note that some mixing-and-matching is possible here, and that the actual limit is six *affiliations* rather than six authors. For example, the following is perfectly reasonable:

```
\twoauthors{First Author}{Second Author and Third Author}
\twoaffiliations{Affiliation of First Author}{Affiliation of Second and Third Authors}
```

and will result in the affiliation of the first author being typeset below the first author's name, and the affiliation of the second and third author's being typeset below their names.

2.2 Title and header information

There are a total of four title and header related fields that can be specified. Note that if you do not specify all the fields the information from the fields you do specify will be used in place of the missing fields. The four fields are:

```
\title{}
\shorttitle{}
\rightheader{}
\leftheader{}
```

Note that there is some possibility for confusion here, as these commands are labeled according the where the information will go in the *published* version of the paper and *not* in the manuscript version. Thus the shorttitle command actually contains the information that will go in the *right header* of the manuscript, while the **\rightheader** command contains the information that will be printed as the *running head* in the manuscript. The **\leftheader** command contains information to be printed in the left header of the published manuscript (usually the author's name) and is not used in manuscript mode. Finally, the **\title** command contains the full title of the article.

2.3 Headings and heading levels

The APA manual specifies different heading styles depending on the total number of headings in a manuscript. In order for LaTeX to figure out how to set your headings, you therefore need to tell it how many headings you have. This is done by issuing the **\headinglevels**{} command in your preamble. Note that the number of headings is specified by *verbal* rather than *numeric* arguments. Thus if your manuscript has four headings, you need to issue the command **\headinglevels**{four}, *not* **\headinglevels**{4}. The manual specifies heading schemes for up to five heading levels.

Once you have specified the number of heading levels in the preamble, you issue section commands as follows:

```
\section{}
\subsection{}
```

```
\subsubsection{}
\paragraph{}
\subparagraph{}
```

Note that all of these may or may not be used, depending on the number of headings. For example, if you have only three heading levels, you should use only the \section, \subsection, and \subsubsection headings. If you decide to change the number of heading levels, LaTeX will automatically adjust the appearance of the headings appropriately.

2.4 Enumeration and seriation

Itemized and enumerated lists can appear in APA style manuscripts, but their use is not common. To create bulleted or numbered lists you can simply use the standard itemize and enumerate environments. More commonly in APA style documents, one wishes to have itemized lists that appear in the body of the text, like (a) this, (b) that, and (c) the other thing. This can be accomplished using the seriate environment, as in \begin{seriate} \item this, \item that, and \item the other thing \end{seriate} and has the advantage of being automatically relabeled if you change the order.

2.5 Cross-references, in-text citations, and reference lists

Cross referencing in LaTeX is easy and very useful. Basic cross referencing can be implemented using just two commands. In order to refer to something later on in your document, you need to give it a label, as in \label{ex:label}. Then you can refer to it by it's label, as in \ref{ex:label}. This is especially useful for tables and figures, because if you change the order, the in-text references to them will automatically be updated. You can also label sections, equations, and footnotes.

For citations, `apa.cls` uses the `apacite` package, which makes formatting citations a very easy process. For the most part, citations in APA style should be very familiar to BibTeX users. For example, assuming I have an entry for the present work in my `.bib` file with the cite-key `Zahn2008`, I can cite it by writing \cite{Zahn2008}. This will result in (Zahn, 2008) appearing in the text of the document, and the reference will be automatically inserted into the reference list at the end of the document. For details on the `apacite` package, refer to the manual[5].

Using BibTeX is very simple, and it works great. I've used several different reference management software, and found most of them to be complicated and/or buggy. BibTeX on the other hand is easy to use and it just works.

3 Installing and setting up Sweave

The preceding sections focused on features of LaTeX that I find particularly useful. In this section, I show how the usefulness of LaTeX can be further enhanced by combining it with R.

The documentation accompanying R and various LaTeX distributions covers the installation and configuration of these programs quite well, and so I will not discuss their installation here. However, having these programs installed and working properly is a pre-requisite for setting up Sweave, so make sure you have both working on your computer before following the instructions below[2].

I prefer to run Sweave from within a text editor (I use TextMate on OS X), and therefore these instructions are geared towards setting things up this way. If you prefer to run Sweave from the command line, simply disregard the parts about configuring your text editor.

3.1 Setting up Sweave on a Windows computer

In the following I assume you are using TeXmaker with the MikTeX distribution of LaTeX, but the general procedure should be very similar with alternative LaTeX distributions/front ends.

Sweave comes with the standard distribution of R, and can be run from within the R program without any additional configuration. However, I prefer to run Sweave directly (i.e., from outside of R), and this requires a few extra steps. First, you will need to install a batch file[3] so you can call Sweave from your text editor. To install the file, you can either place it somewhere in your PATH, or simply place it anywhere you like and add the directory to your PATH.

At this point you should be able to run Sweave from the command line. If you prefer to call it directly from within your text editor, you will need to tell your text editor where to find the `sweave.bat` file. Open TeXmaker, and select `Edit` → `Preferences`. Next, select a function you won't need (I recommend `DVI viewer`), replace the command with sweave.bat −−pdf *%.Rnw*, and click `OK`.

[2] I am currently working on a complete Sweave guide that (among other things) details the installation and set up of Sweave assuming no previous experience with either LaTeXor R. Please contact me at the e-mail address listed at the beginning of this document to obtain this guide.

[3] You can download the file from http://cran.r-project.org/contrib/extra/batchfiles/.

On Windows, it is common to encounter difficulties because LaTeX cannot find the Sweave.sty file. To correct this, copy the sweave.sty file from C:\Program Files\R\R-2.6.2\share\texmf and paste it somewhere your LaTeXprogram can find it.

3.2 Setting up a Sweave on Mac OS X

The procedures involved in setting up a Mac OS X workstation are even easier than those required for setting up a Windows workstation. If you want to be able to call Sweave outside of R, you will need to install a shell script[4]. To install the script, copy it to /usr/local/bin, then open the Terminal program and type sudo chmod +x /usr/local/bin/Sweave.sh to make it executable.

At this point you should be able to call Sweave from the command line. If you want to call Sweave from within TEXshop you need to tell TEXShop where to find the Sweave.sh shell script. Open TEXShop and click TEXShop → Preferences. Click the Misc tab and type /usr/local/bin/Sweave.sh –ld in the LaTeX Progam Personal Script field.

3.3 Setting up Sweave on Linux

The procedures involved in setting up a Linux workstation are also very straightforward, although the details will vary depending on which distribution and text editor you are using. The present example is based on Kubuntu Linux, using Kile.

If you want to be able to call Sweave outside of R, you will need to install a shell script (see footnote 4). To install the script, copy it to /usr/local/bin, then open the Konsole program and type sudo chmod +x /usr/local/bin/Sweave.sh to make it executable.

Next, you may want to tell Kile where to find the Sweave.sh shell script. Open Kile and click Settings → Configure Kile. Click the Tools tab on the left-hand side of the preferences window, and select Build. Click the New Tool button at the bottom of the preferences window. Name the new tool Sweave, click next, and then Finish. In the resulting screen, type Sweave.sh in the top box, and –ld `\%*source*' in the bottom box.

[4] You can download the Sweave.sh script from http://cran.r-project.org/contrib/extra/scripts/.

4 Sweave basics

In this section I give an overview of and basic introduction to Sweave. As you will soon find, it is not difficult to learn, but it does require that you have some knowledge of both R and LaTeX. To create a Sweave document, simply set up your LaTeX preamble as usual, use the `Sweave` package, and save the file with a `.Rnw` extension. For example, the first two lines of the present article read

```
\documentclass{pracjourn}
\usepackage{Sweave}
```

and the file is named `LearningToSweaveAPAStyle.Rnw`.

4.1 What happens when you run `Sweave`?

When you run `Sweave`, the output is a regular LaTeX file, with the R code and results included. If you have your system set up as described above, this is all hidden from view, because the program is set up to automatically run LaTeX on the Sweave output. But what looks like a single seamless process is actually a two-stage affair. This is useful in an important way, namely that we can run `Sweave`, and then edit the resulting `.tex` file to make any manual changes we wish to the results returned from R.

4.2 Details about `Sweave` syntax

Unlike LaTeX and R, `Sweave` is not extensive or complicated. This is because `Sweave` has a relatively simple purpose: it allows you to combine LaTeX and R code into a single document. The basic Sweave syntax consists simply of a << >>= marker indicating that we want to begin an R code section, followed by one or more R commands, followed by a @ marker indicating you are done writing R commands and wish to switch back to LaTeX. Although this basic syntax is extremely simple, it's actually slightly more complicated than this, because you can put option commands inside the openning marker. In the following, I discuss only a few of the more commonly used options; for a full description the available options, see the Sweave user manual[4].

In the examples that follow, I use a simple contrived data set that takes the following form:

```
<<echo=false>>=
Participant <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

```
Condition <- factor(c(1, 1, 1, 1, 1, 2, 2, 2, 2, 2))
Score <- c(4, 3, 5, 4, 4, 2, 2, 6, 5, 6)
Data <- data.frame(Participant, Condition, Score)
Data
@
```

These R commands create the following simple data frame:

```
   Participant Condition Score
1            1         1     4
2            2         1     3
3            3         1     5
4            4         1     4
5            5         1     4
6            6         2     2
7            7         2     2
8            8         2     6
9            9         2     5
10          10         2     6
```

If you don't understand what these commands are doing, make sure to consult an introductory R guide[5] before proceeding.

4.3 The echo option

The *echo* option dictates whether the R *input* is incorporated into the typeset document. For example, if we want to calculate the grand mean of the `Score` variable for the `Data` dataset created above, we could say

```
<<echo=true>>=
mean(Score)
@
```

which gives us

```
> mean(Score)

[1] 4.1
```

or, we could say

```
<<echo=false>>=
mean(Score)
@
```

which gives

```
[1] 4.1
```

Notice that in the first case the command returned both the command and the result, while in the second case only the result appears.

[5] Many excellent and free guides are available at http://www.r-project.org/.

4.4 The results option

The *results* option tells Sweave how we want the results formatted. We can hide the results completely, as in

```
<<echo=false,results=hide>>=
mean(Score)
@
```

We can return the results already formated in LATEX code (useful especially when we have R create LATEX tables for us; more on this later), as in

```
<<echo=false,results=tex>>=
mean(Score)
@
```

or we can have the result returned verbatim (the default), as in

```
<<echo=false,results=verbatim>>=
mean(Score)
@
```

The reader is encouraged to play around with these options in order to see what they do first-hand.

5 Including results from R into your document

The main purpose of Sweave is to allow you to perform statistical analyses and include the results directly into your LATEX document, without the need to switch programs, copy-and-paste, or re-type results. In this section I provide examples illustrating how to do this. Many of the examples use the *Prestige* dataset from the `car`[1] R package. The Prestige data set contains six variables, as described in: the `car` package documentation:

education Average education of occupational incumbents, years, in 1971.

income Average income of incumbents, dollars, in 1971.

women Percentage of incumbents who are women.

prestige Pineo-Porter prestige score for occupation, from a social survey conducted in the mid-1960s.

census Canadian Census occupational code.

type Type of occupation. A factor with levels (note: out of order): bc, Blue Collar; prof, Professional, Managerial, and Technical; wc, White Collar.

In the following I'll present a number of examples using these data. First, I illustrate how to calculate statistics and included them in the text of you document.

Next, I'll show you how to make R create LaTeX tables for you. Finally, I'll show you how to create graphics in R and incorporate them into your LaTeX document.

5.1 Incorporating R results using the Sexpr command

There is another way to insert short chunks of R code into your LaTeX document. Quite simply, you can issue the `\Sexpr{}` command, and put your R code inside the brackets.

For example, if I say `\Sexpr{2+2}` I will get 4. The nice thing about this command is that the indexes (i.e., those pesky `[1]`'s peppered throughout the examples above) are automatically suppressed, and the result is returned in the same font as the rest of the normal text in your document.

The `\Sexpr` command can be very useful. Instead of copying and pasting results from SPSS into Word, we can do all our calculations right in our LaTeX document. Here is a brief example:

```
The average income was
\Sexpr{mean(income)}, with a standard deviation of \Sexpr{sd(income)}.
The average level of education was
\Sexpr{mean(education)}, with a standard deviation of
\Sexpr{sd(education)}. The correlation between income and education was
\Sexpr{cor(income,education)}.
```

This will be typeset as:

> The mean income was 6798, with a standard deviation of 4246. The mean level of education was 11, with a standard deviation of 2.7. The correlation between income and education was 0.58.

Once I got used to reporting results this way, it's hard for me to imagine how I ever worked without it. It's so simple, so easy, that I can't help but wonder why anyone would go to all the trouble of having separate interfaces to their statistical and document preparation software. And it gets even better with Sweave, as we can do exactly the same kind of operations with tables and figures that we just did with single values.

5.2 Including tables in your document with `xtable`

There are two separate R libraries that can be used to create LaTeX tables in R. The first is the `xtable` package[8], which was designed exclusively for this purpose. The other is the `Hmisc` package[3], which includes the `latex()` command. I'm going to illustrate the use of `xtable` because I've found it easier to use. `xtable` formats table results in LaTeX style, turning the utilitarian R output into nicely formated tables. To load the `xtable` package, simply write `library(xtable)` inside a Sweave tag.

Once the libraries are loaded, you are almost ready to begin making tables. To make a table of descriptive statistics we need to do a bit of work in R before calling `xtable`. Specifically, we need our descriptive statistics to be formatted in a *data frame*, after which we can run `xtable` on the data frame. Here is an example that creates a table displaying descriptive statistics for the `income` and `education` variables found in the `Prestige` dataset:

```
<<echo=false ,results=tex>>=
means <- c(mean(income), mean(education))
sds <- c(sd(income), sd(education))
mins <- c(min(income), min(education))
maxs <- c(max(income), max(education))
descriptives <- data.frame(means, sds, mins, maxs,
row.names =c(”Income”, ”Education”))
names(descriptives) <- c(”Mean”, ”SD”, ”Min”, ”Max”)
descriptives.table <- xtable(descriptives, caption =
”Descriptive␣Statistics”, label = ”tab:descriptive2”)
print(descriptives.table, include.rownames=TRUE)
```

This will produce the result displayed in Table 1.

Table 1: Descriptive Statistics

	Mean	SD	Min	Max
Income	6797.90	4245.92	611.00	25879.00
Education	10.74	2.73	6.38	15.97

The `xtable` package also has extensive methods for creating tables displaying the results of inferential statistics. For example, we can create an ANOVA table summarizing the results of a linear regression model predicting `income` from `education`, as follows:

```
<<echo=false , results=hide>>=
library(xtable)
```

```
xtable(anova(lm(income ~ education)),caption =
"ANOVA␣Table", label= "tab:anovatable1")
@
```

The results are displayed in Table 2. This basic method can be applied to just about any model object. About the only thing that you might object to is the somewhat unconventional labels used across the top of the table. It is possible to modify these values in the model object before calling xtable, but I find that this is often more trouble than it's worth, and I'm usually better off just using xtable to get an approximate result of what I want, and then tweaking the table (by editing the LaTeX file that Sweave produces) to get the details right.

Table 2: ANOVA Table

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
education	1.00	607421386.02	607421386.02	50.06	0.00
Residuals	100.00	1213392025.00	12133920.25		

To do this, simply open up the file, and find the table generated by `xtable`. In the present case, we find

```
% latex table generated in R 2.6.2 by xtable 1.5-2 package
% Fri Mar 28 13:02:56 2008
\begin{table}[ht]
\begin{center}
\caption{ANOVA Table}
\label{tab:anovatable1}
\begin{tabular}{lrrrrr}
  \hline
 & Df & Sum Sq & Mean Sq & F value & Pr($>$F) \\
  \hline
education & 1 & 607421386.02 & 607421386.02 & 50.06 & 0.0000 \\
  Residuals & 100 & 1213392025.00 & 12133920.25 & & \\
   \hline
\end{tabular}
\end{center}
\end{table}
```

Now all we need to do is edit the table to make it look exactly like we want it to, and re-typeset the document.

5.3 Graphics in Sweave documents

The beauty of Sweave is that we can use R to create the graphics, and include the results all from a single, unified, and consistent interface (i.e., our text editor). In the following sections, I will walk you through a way of doing exactly this. R comes with robust plotting capabilities, and there are several add-on packages that extend these capabilities. I'm going to use the `ggplot2` package[7] in the following examples, but the general procedure should work even if you use another plotting package.

To generate a figure, we simply place the required R code inside the <<>> = ... @ tags. There are actually a few different ways to go about inserting the figure. The method I'm going to present here is specifically chosen because it makes it easier to produce figures that are sized appropriately, with appropriately sized axis labels and legends (something that can be tricky with other methods). Specifically, I recommend creating your graphic and saving it to a `.pdf` file, and then inserting the image into a figure environment. For example, we can easily represent the relations among `income`, `education`, `type` of occupation, percentage of `woman` in a field, and the `prestige` of a field, all in one information-dense graphic. The code for doing all this is simply

```
<<echo=false, results=hide>>=
pdf(file=”splot12.pdf”, width=6, height=5)
qplot(education,income, shape=type, size=women, colour=prestige,
xlab=”Education”, ylab=”Income”)
dev.off()
@
\begin{figure}
\centering
\includegraphics[width=6in, height=5in]{splot12.pdf}
\caption{A scatterplot displaying the relationships
among Income, Education, Prestige, Type of occupation,
and percentage of Women in the field.}
\label{fig:fig12}
\end{figure}
```

The result is displayed in Figure 1, and really shows off the impressive capabilities of qplot().

The R command pdf() tells R to save the graph to a `.pdf` file. Then we include the graph with the standard \includegraphics[]{} command making sure to specify the same size in the options as we did in the options to the pdf() command. Note that the pdf() command expects the units to be in inches, so just write the number of inches you want, while the \includegraphics[]{} command needs to be told that the values are in inches (i.e., write \includegraphics[width = 3in]{**file**} rather than just \includegraphics[width=3]{**file**}, which will make LaTeX complain that you didn't tell it what to do).

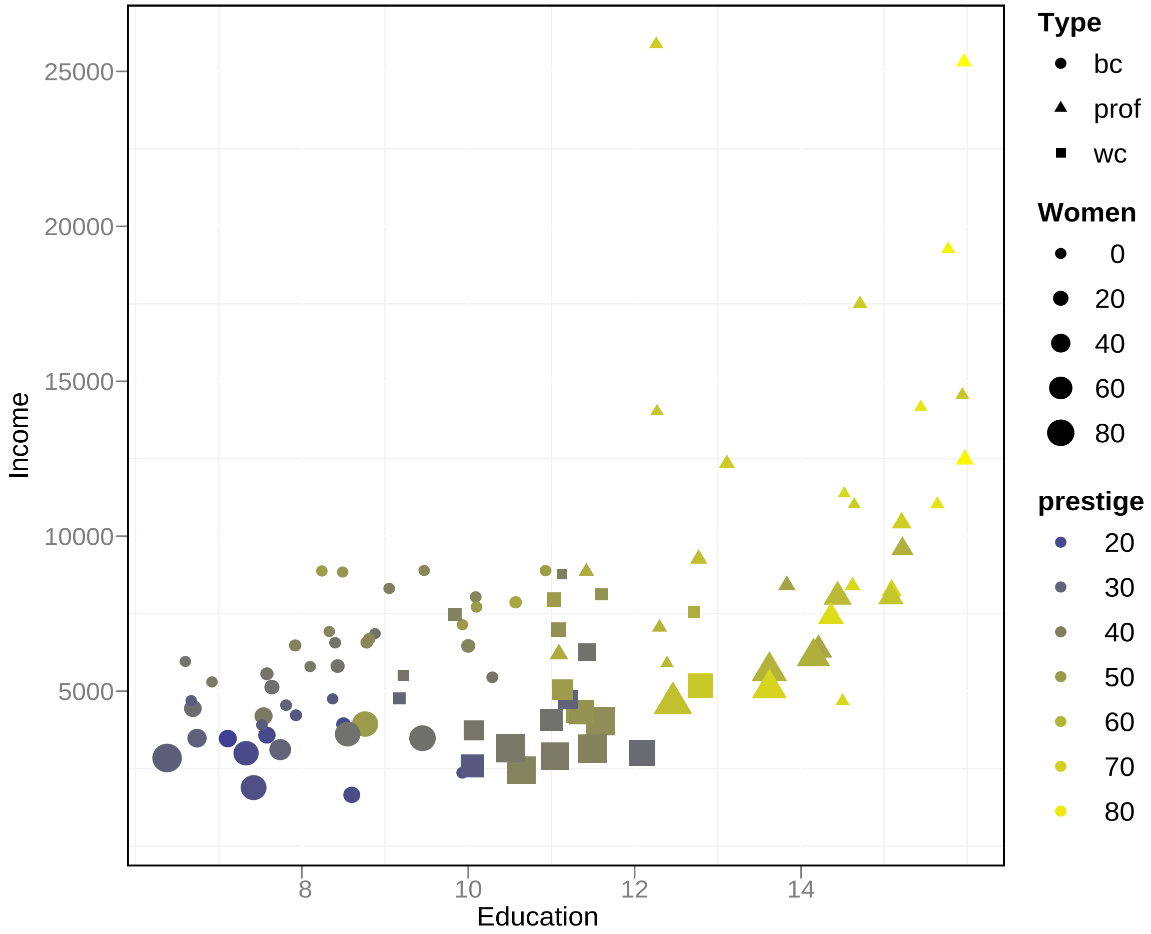

Figure 1: A scatterplot displaying the relationships among Income, Education, Prestige, Type of occupation, and percentage of Women in the field.

6 Collaborating with those who don't use Sweave

One of the most frustrating things about learning to use Sweave is that it can be difficult to collaborate with colleagues who do not. A related concern is that journals will not accept articles in LaTeX format. Both of these are valid concerns. They can be worked around, but in my experience there is no completely satisfying solution.

There are a number of programs that can convert simple LaTeX files to `.rtf` or `.html`, both of which can be read by most word processors, including Microsoft

Word. Such programs include latex2rtf and latex2html, and others[6]. Unfortunately, my experience is that such converters do not handle APA formated LATEX documents well, if they can even be coaxed into producing any output at all.

As strange as it sounds, my advice is to forget about converting your LATEX document to a Word-readable file directly. Instead, typeset the document as a `.pdf` file, and then convert the `.pdf` file to `.html` (`.html` files can be read by most modern word processors, including Microsoft Word). There are stand-alone conversion programs that will do this for you, such as the excellent pdftohtml[7]. If you have Gmail you can send the `.pdf` document to yourself, and click the `View as HTML` button in the `Gmail` web interface. Finally, Adobe has an on-line converter[8] that can convert `.pdf`'s to `.html`, although this can be slow for long documents. Once you have the `.html` formated document, open it in your word processor and fix the formatting (which will usually be close but not exactly right), and save it in `.doc` format.

The conversion to `.doc` format will usually be easier if you set your LATEX document to left-justified (aka 'ragged right'; **\raggedright**), and if the margin settings in LATEX match the margin settings in your word processor. If you use `apa.cls` in manuscript mode you shouldn't have too much trouble.

7 Conclusions and final thoughts

The preceding sections document some of the things I've learned to do with LATEX and R. Some sections descended into detailed discussions of various techniques, taking us a long way from the general, largely personal tone of the opening section. In this final section, I want to step back from the discussion of specific techniques, and offer my personal thoughts about using Sweave.

In the summer of 2006, I was looking for a calculator program for OS X. I had trouble finding one I liked, and eventually started using R as a simple calculator. I can almost hear the R developers shrieking in horror to think that someone would use their powerful and sophisticated program as a *calculator*, but the truth is that R serves this function very well. As time went on, I started using R for analysis of variance and linear regression as well. Over the course of several months, I gradually began using R more and more, and SPSS less and less.

[6] A list of conversion programs is available at http://www.tug.org/utilities/texconv/textopc.html

[7] Available at http://sourceforge.net/projects/pdftohtml/

[8] Available at http://www.adobe.com/products/acrobat/access_onlinetools.html

In the summer of 2007, I taught an introductory statistics course. I needed to prepare lecture slides that included a lot of math, and Google pointed me towards LaTeXit, a small OS X program for typesetting math, based on LaTeX. I was so impressed with LaTeXit, that I started teaching myself to use LaTeX. It didn't take long for me to find out about Sweave, and since that time I've used LaTeXand R for all my papers, as well as my presentations (using the `Beamer` package).

In this way, I found myself using tools that are completely different from those used by the majority of my colleagues, and different from the tools I've learned to use since grade school. In general, my impressions so far are that (a) LaTeX is both easier to use and produces better output than Word, and (b) R is easier to use and produces better output that SPSS. Of course neither of those statements were true when I first started. Indeed, I almost gave up on R at one point because it seemed more difficult than SPSS. But once I took a few months to learn it, it became quite natural and easy to use.

Overall then, I am quite happy to be the only one in my department using LaTeX and R. I don't have to pay expensive licensing fees, and I get to take advantage of all the wonderful features of LaTeX and friends. In addition, I've learned how to convert `.pdf`'s to `.html`, which means I can still collaborate effectively with Word users, and I can always convert my documents before submitting them for publication if I have to. In sum, my experience with LaTeXand R have been very rewarding. I hope that by sharing these experiences and giving a few pointers I have inspired you to give Sweave a try.

References

[1] John Fox. *car: Companion to Applied Regression*, 2007. R package version 1.2-7. I am grateful to many people for various suggestions and contributions.

[2] John Fox and with contributions from many others. *Rcmdr: R Commander*, 2008. R package version 1.3-12.

[3] Frank E Harrell Jr and with contributions from many other users. *Hmisc: Harrell Miscellaneous*, 2007.

[4] Friedrich Leisch. *Sweave User Manual*, 2007.

[5] Erik Meijer. *The apacite package: Citation and reference list with LATEX and BibT EX according to the rules of the American Psychological Association*, 2005.

[6] Athanassios Protopapas. *APA LaTex Class File*, 1.3.2 edition, 2006.

[7] Hadley Wickham. *The ggplot book*, 2007.

[8] David B. Dahl with contributions from many others. *xtable: Export tables to LaTeX or HTML*, 2007. R package version 1.5-2.