Getting Nice LaTeX tables out of R

Paul E. Johnson

Feb. 10, 2009

I learned that the R package "memisc" offers a wealth of functions that will facilitate social science research presentations. I was drawn to that by a post in r-help that described its facilities to generate LaTeX tables for statistical reports.

Before I knew of memisc, I had used other techniques to generate LaTeX tables in R. In the Hmisc package, there is a latex function that generates some tables, and I also wrote an R function "outreg" to generate regression tables. A Washington University graduate student recently created an R package called "apsrtable" that also makes regression tables and has functionality almost exactly the same as my outreg. Neither outreg nor apsrtable have the all-encompassing power of memisc, and that is either a plus or minus, depending on your needs & opinions.

To see for myself, I tried to generate a table with memisc's functions. That package provides a nice table generating function genTable that uses a pleasant, simple syntax `genTable(y~x, data=z)`. The memisc package includes a "percent" function and the table generator allows a usage like this: `genTable(percent(y) ~x, data=z)`. To convert rows to columns in accord with my Iron Law of Crosstabs, `t()` can be wrapped around the genTable command. If one wants LaTeX output, one can either do `toLatex(genTable(percent(y) ~x, data=z))` or in two steps,

```
mytable <- genTable(percent(y) ~x, data=z)
toLatex(mytable)
```

I copied and pasted the LaTeX table into a document, but the LaTeX processor could not understand some commands. That sent me off on a little goose chase.

The table-making standards are spelled out in a few places. The Essay "Publication quality tables in LaTeX" can be found online at `http://www.ctan.org/tex-archive/macros/latex/contrib/booktabs/booktabs.pdf`. As far as I can tell, it was written as a part of the development for a LaTex package called "booktabs" that is intended to facilitate professional looking tables in LaTeX documents. Another essay by Lapo Fillipo Mori, called "Tables in LaTeX 2_ε : Packages and Methods" offers a broader overview (obtain that at:`http://www.tug.org/pracjourn/2007-1/mori/mori.pdf`

Here's a table that comes from Hmisc "latex" function with the file option set to empty, so the results display on the screen (latex(x, file="")).

```
\begin{center}
\begin{tabular}{lrr}\hline\hline
\multicolumn{1}{l}{with}&
\multicolumn{1}{c}{F}&
\multicolumn{1}{c}{M}
\\ \hline
C&$ 8$&$12$\\
T&$22$&$18$\\
\hline
\end{tabular}
\end{center}
```

That generates a table that looks like this:

with	F	M
C	8	12
T	22	18

That table does not use some of the special table magic that the publishing experts want us to use. It will "work" in a basic LaTeX document, but it is not optimal because it uses \hline to provide lines and it does not have much

So I consider instead using memisc's toLatex function. Here's the syntax that appears in R from "toLatex"

```
\begin{tabular}{lD{.}{.}{0}D{.}{.}{0}}
\toprule
 & \multicolumn{1}{c}{F} & \multicolumn{1}{c}{M} \\
\midrule
C & 8 & 12 \\
T & 22 & 18 \\
\bottomrule
\end{tabular}
```

That drives LaTeX insane when I try to view the document. There are many errors about illegal characters, illegal array specifiers (it does not like the "D" markers in the tabular command) and then errors about unrecognized macros like \toprule and \midrule.

After fiddling about a while, I learned that the LaTeX that comes out of the memisc package has some specific formatting assumptions. If you don't have the required packages in your LaTeX document preamble, then you will see errors when you try to view the document.

Step 1. In the document preamble, use either "\usepackage{ctable}" or "\usepackage{booktabs}". These packages provide the table separators \toprule, \midrule, and \bottomrule. (I don' know the difference between ctable and booktabs, both seem to work. It seems the latter may be more popular.)

Step 2. The preamble needs "\usepackage{dcolumn}".

If you don't want to use "dcolumn", you can just replace that complicated looking D{}{} stuff with either "l", "c", or "r" (for "left", "center", or "right" aligned). This modified tabular object does work:

```
\begin{center}
\begin{tabular}{lrr}
\toprule
 & \multicolumn{1}{c}{F} & \multicolumn{1}{c}{M} \\
\midrule
C & 8 & 12 \\
T & 22 & 18 \\
\bottomrule
\end{tabular}
\end{center}
```

That generates the following table.

	F	M
C	8	12
T	22	18

However, it is not necessary to do that if you have the LaTeX package "dcolumn". After inserting the preamble option \usepackage{dcolumn}, the output from memisc works!

	F	M
C	8	12
T	22	18

Furthermore, we can get a nice looking percentage table from the memisc output as well. This LaTeX ma

```
\begin{tabular}{lD{.}{.}{0}D{.}{.}{0}}
\toprule
 & \multicolumn{1}{c}{F} & \multicolumn{1}{c}{M} \\
\midrule
C & 27 & 40 \\
T & 73 & 60 \\
N & 30 & 30 \\
\bottomrule
\end{tabular}
```

Leads to a table object that looks like this;

	F	M
C	27	40
T	73	60
N	30	30

I googled "multicolumn latex" for 2 minutes and understood how to modify this so that the column variables match my desire and I insert \% for percent signs;

	Respondent Sex	
Classification	F	M
C	27%	40%
T	73%	60%
N	30	30

I might even think this makes more pleasant output, but I'm not sure. It is a bit of a hassle because it is necessary to insert another column (note I insert an "l" in the tabular).

		Respondent Sex	
		F	M
Classification	Control	27%	40%
	Test	73%	60%
	N.of Cases	30	30

Here's the R code I was using to generate this example output

```
 library(UsingR)
## Ordinary table syntax wants the row variable first, column second
## So, if we are thinking of "control" in the reaction.time as the output
with(reaction.time, table( control, gender) )
library (Hmisc)
mt <- with(reaction.time, table( control, gender))
prop.table(addmargins(mt,1))
latex(addmargins(mt,1) file="")
library(memisc)
genTable(gender~control, data=reaction.time)
genTable(percent(gender)~control, data=reaction.time)
## Maybe you view "control" as the output? want N: toLatex(genTable(control~
    gender, data=reaction.time))
## want percents;
toLatex(genTable(percent(control)~gender, data=reaction.time))
```