September 29, 2016

1 Statistical Reporting

Statisticians write many reports for a variety of purposes, so it helps to have a good system to do this. There are some R packages designed for this, but it is fairly straightforward to create dynamic reporting mechanisms with R.

1.1 Simple tables

Creating files that hold figures is supported by R through the `pdf`, `jpeg` and related functions. The main hurdle is getting high resolution figures (using the `res` argument to `jpeg` for example): to do so one must change the default height and width arguments. The trick here is to make them large (like 2000). Tables are not as simple, but by using the text editor latex one can create publication quality tables with a little work. First put all of your summaries into a matrix, then provide row and column names. One can then use the `xtable` command and R will produce the latex code to generate the table using latex. Here is an example.

```
> library(xtable)
> tab=matrix(NA,2,3)
> tab[1,]=c(quantile(faithful[,1],.25),median(faithful[,1]),quantile(faithful[,1],.75))
> tab[2,]=c(quantile(faithful[,2],.25),median(faithful[,2]),quantile(faithful[,2],.75))
> rownames(tab)=names(faithful)
> colnames(tab)=c("First Quartile", "Median", "Third Quartile")

> xtable(tab)
% latex table generated in R 3.2.3 by xtable 1.8-2 package
% Thu Sep 29 10:30:13 2016
\begin{table}[ht]
\centering
\begin{tabular}{rrrr}
  \hline
 & First Quartile & Median & Third Quartile \\
  \hline
eruptions & 2.16 & 4.00 & 4.45 \\
  waiting & 58.00 & 76.00 & 82.00 \\
   \hline
\end{tabular}
\end{table}
```

One could then put this text into a text file with the appropriate header, put the statement

```
\end{document}
```

at the end, then run latex on that file to produce an output file with the desired table. I use a variant called `pdflatex` that one runs at the linux command line as follows (here I assume the text file I have created is called example.tex)

```
$ pdflatex example.tex
```

and that produces a pdf file called `example.pdf` that has the table. This can then be inserted into other documents, or combined with other pdf documents with a command line tool like `pdftk`.

A better way than writing the latex code to your R terminal then cutting and pasting into an external file is to write the table to the external file, then use the latex command `input` to include the table in your latex code. To do that we use the `print` command

```
print(xtable(tab,caption=c("Summary of old faithful data.","Summary of old faithful data."),
  label="table1",digits=2), file="table1.tex", table.placement="H",
  sanitize.colnames.function=function(x){x})
```

then put

```
\input table1.tex
```

in your latex file where you want the table. You can read the help for `xtable` and experiment to find out more about this.

So what is the appropriate header? That depends but I have a link to a file called `latexHeader.tex` on this site that has a lot of the packages that I find helpful. Look up the package names to find out about what these packages can do.

1.2 Simple reports

Suppose I want to actually create a report with multiple figures and tables? The simplest thing is to start with a latex header file as above, but then add the text necessary to include figures and tables as you generate your report. For example, the following 2 R functions will write the necessary latex code to your file:

```
makeFig=function(figfile,texfile,caption,label){
  chvec=rep(NA,6)
  chvec[1]="\\begin{figure}[H]"
  chvec[2]="  \\centering"
  chvec[3]=paste("    \\includegraphics{",figfile,"}",sep="")
  chvec[4]=paste("  \\caption{",caption,"}",sep="")
  chvec[5]=paste("  \\label{",label,"}",sep="")
  chvec[6]="\\end{figure}\n"
  write.table(chvec,file=texfile,quote=F,row.names=F,col.names=F,append=T)
}

makeTable=function(tablefile,texfile){
  write.table(paste("\\input{",tablefile,"}\n",sep=""),file=texfile,quote=F,row.names=F,
  col.names=F,append=T)
}
```

To use these functions you specify the name of the latex file (i.e. `texfile`), the name of file that holds the table or figure, and for figures you supply a caption and a label (these were added to our table when we called `xtable`). You can also dump text into your report, for example, the following would start a new subsection in the latex file `eyeReport.tex`

```
write.table("\\subsection{Detailed Eye Survey}\n",file="eyeReport.tex",quote=F,row.names=F,
  col.names=F,append=T)
```

Here is an example of creating a table for a report

```
print(xtable(t1,caption=c("Number of participants enrolled and having eye exams.",
  "Number of participants enrolled and having eye exams."), label="table1",digits=0),
  file="table1.tex", table.placement="H",sanitize.colnames.function=function(x){x})

makeTable("table1","eyeReport.tex")
```

and here is a figure

```
jpeg("presenting-dVA.jpg",height=2000,width=2000,res=300)
boxplot(exp(y)~surv1Lab,ylab="dVA: Geometric mean across eyes")
dev.off()

makeFig("presenting-dVA.jpg","eyeReport.tex",caption="Presenting dVA",label="presenting-dVA")
```

So if you use the `xtable` package in conjunction with a couple of very simple functions for writing text to an external file you can create files with a collection of R code that analyzes data and adds to an existing file text that includes the figures and tables that you create. Remember to add

```
write.table("\\end{document}",file="eyeReport.tex",quote=F,row.names=F,col.names=F,append=T)
```

at the end of your command file or you will get frustrating error messages when you try to compile your latex code. If you don't like the appearance of the resulting file you can alter your R code and rerun or simply edit the `.tex` file directly.