

Sweave markup

To ensure that comments are retained in your code, place the following in your document preamble (actually it does not have to be in the preamble).

```
\\SweaveOpts{keep.source=TRUE}
```

Setup at the beginning

```
<<setup, echo=F>>=
opt <- options(SweaveHooks=list(fig=function(){par(cex=0.8,
          mar=c(4.1,3.6,1.6,1.1), pty="s", lwd=1,
          mgp=c(2.25,0.5,0), tck=-0.025)}))
pdf.options(pointsize=8)
@%
```

You can also have `options(prompt=" ", continue=" ")`, or `options(continue=" ")` If you want the first of these options, use it once everything is working. Also, include near the end of your document:

```
<<prompt, eval=t, echo=f>>=
options(prompt="> ", continue="+ ")
##
@%
```

If your document does not process properly through Sweave, and you have turned off the prompt and continuation prompt, you can restore it by typing at the command line:

```
options(prompt="> ", continue="+ ")
```

Create named code fragment, use, execute

For example

```
%% Create named code fragment
<<countzeros, eval=f, echo=f>>=
countzeros <- function(x)sum(!is.na(x) & x==0)
aggregate(allsets[, c("re74", "re75", "re78")], list(group=allsets$trt),
          FUN=countzeros)
@%
```

```
%% Execute code fragment, do not echo
<<do-countzeros, echo=f, eval=t>>=
<<countzeros>>
@%
```

```
%% Echo code
<<code-countzeros, echo=t, eval=f>>=
<<countzeros>>
@%
```

Use the argument `results=hide`, if you wish to hide any results from the calculation.

Graphs

Here is an example

```
%% Create named code fragment (here called 'hist')
<<hist, echo=f, eval=f>>=
library(lattice)
print(histogram(~educ | trt, data=allsets, layout=c(4,1)))
@%
```

```
%% Create graph
\setkeys{Gin}{width=\linewidth}
\begin{figure}
<<do-hist, fig=t, width=8, height=3.75, echo=f>>=
<<hist>>
@ %
\caption{Histograms, used to compare distributions of Years of education.}
\end{figure}
```

Note that @ % in place of @ is optional. The % tells editors to respect the line feed when reformatting.