Math3346 Assignment 1 – Solutions and Commentary

Lecturer: John Maindonald

September 3, 2009

General Comments

- Use tables to summarize information. Use text to draw attention to major features in the tables. Place the main focus on features that are clear and unequivocal, with more minor matters consigned to the later brief comment or to footnotes or endnotes.
- In an assignment such as this, most code should be included, properly referenced, at the end of the document. Including limited amounts of code in the text is acceptable, providing it has a strong "literate programming" character!
  [In a report for a manager, omit code from the main document.]
- Round proportions to (in these data) 3 or 4 decimal places.
- Be judicious in the choice of graphs. If you have 10 or 12 graphs spread over 4 or 5 pages, you likely need to find a more succinct form of presentation. Aim at a layout where it will be easy to identify the major differences. Overlaid plots of density estimates sometimes work well, for an audience that has some modest level of technical sophistication.
- Some forms of presentation that can be effective with an audience that has the right level of technically sophistication are not suited to use in reports that are intended for a general audience. Thus in considering use of, for example, density plots and boxplots, keep the audience in mind.
- Where a variable has missing values the total of zeros (or other such total) will be the total of those known to be zero. On the (common) assumption that data are missing at random, division by the total of non-missing values gives an estimate of the proportion of zeros. If data are not missing at random, then the number that results may be biased, perhaps so seriously biased that it is uninterpretable. This seems an issue for the variable `re74` in the present data.

Solutions

1. Write brief notes on each of the columns of data, noting whether columns should be treated as numeric or categorical. [1 mark]
   NB: Describe, where relevant, what codes 0 and 1 mean.
   Document (briefly) each of `countNA()`, `countzeros()` and `countlist()` [2 marks]

2. For each column of the data and for each of the four groups

   (a) Determine the number of missing values. [1 mark]

	age	educ	black	hisp	marr	nodeg	re74	re75	re78
cf-cps1	0	0	0	0	0	0	0	0	0
cf-psid1	0	0	0	0	0	0	0	0	0
exp-ctl	0	0	0	0	0	0	165	0	0
exp-trt	0	0	0	0	0	0	112	0	0

   (b) For each of `re74`, `re75` and `re78`, what number and proportion are zero. [1 mark]

	cf-cps1	cf-psid1	exp-ctl	exp-trt	cf-cps1	cf-psid1	exp-ctl	exp-trt
re74	1913	215	195	131	0.120	0.086	0.750	0.708
re75	1748	249	178	111	0.109	0.100	0.419	0.374
re78	2172	286	129	67	0.136	0.115	0.304	0.226

Note: The proportion of (known) zeros for ***re74*** *has been calculated relative to non-missing data. This is the correct figure if missingness does not carry any information about income. For the present experimental data, this implies a massive change (> 30%, from ∼ 0.4 to > 0.7) in the proportion of non-earners, between 1974 and 1975. As there is no comparable change in the* ***cps1*** *and* ***psid1*** *data, this seems unlikely. Missing values for* ***re74*** *seem then unlikely to be missing at random. The variable* ***re74*** *has therefore been ignored in the subsequent discussion.*

To calculate (known) zeros as a proportion of all data, change ***length(na.omit(x))*** *to* ***length(x)****. [Why would one want this?].*

3. Now examine `re74`, comparing control and treatment data:

   (a) Compare the proportion of `NA`s between the experimental control and treatment. [$\frac{1}{2}$ mark]

```
    0     1
0.388 0.377
```

   (b) Compare the proportion of 0's in each of `re74`, `re75` and `re78` (obviously `NA`s have to be excluded) between the four groups. [$\frac{1}{2}$ mark]

```
     cf-cps1 cf-psid1 exp-ctl exp-trt
re74   0.120    0.086   0.459   0.441
re75   0.109    0.100   0.419   0.374
re78   0.136    0.115   0.304   0.226
```

   (c) For columns that are numeric with more than two unique values, determine for each of the four groups the range of values. [1 mark]

```
          age-l age-u educ-l educ-u re74-l re74-u re75-l re75-u re78-l re78-u
cf-cps1      16    55      0     18      0  25862      0  25244      0  25565
cf-psid1     18    55      0     17      0 137149      0 156653      0 121174
exp-ctl      17    55      3     14      0  39571      0  36941      0  39484
exp-trt      17    49      4     16      0  35040      0  37432      0  60308
```

4. Provide graphs that conveniently summarise differences between the four groups, with respect to `age` and `re75`. [4 marks]

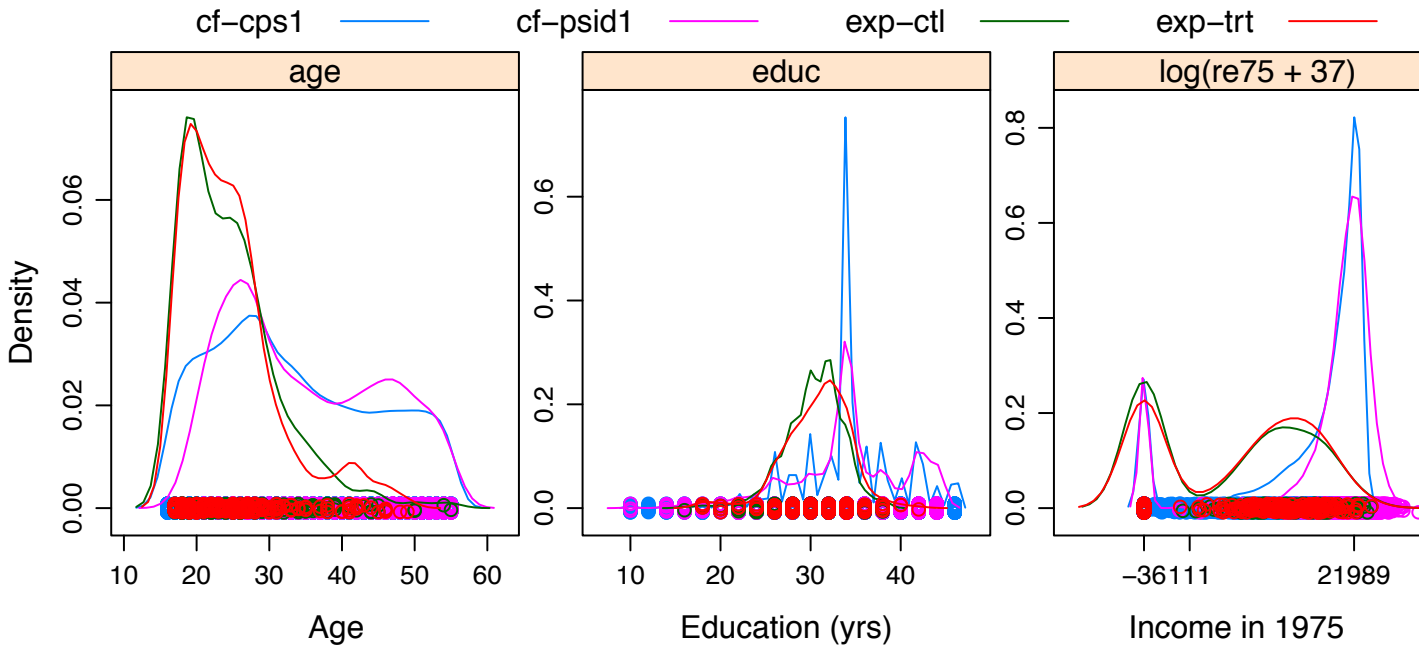

Figure 1: Densityplot comparisons, for Age, for Years of education and for `log(re75+37)`. The logarithmic transformation leads to a more symmetric distribution. The plot for `log(re75+37` has a spike at zero that, in the density plot, is spread either side of zero.

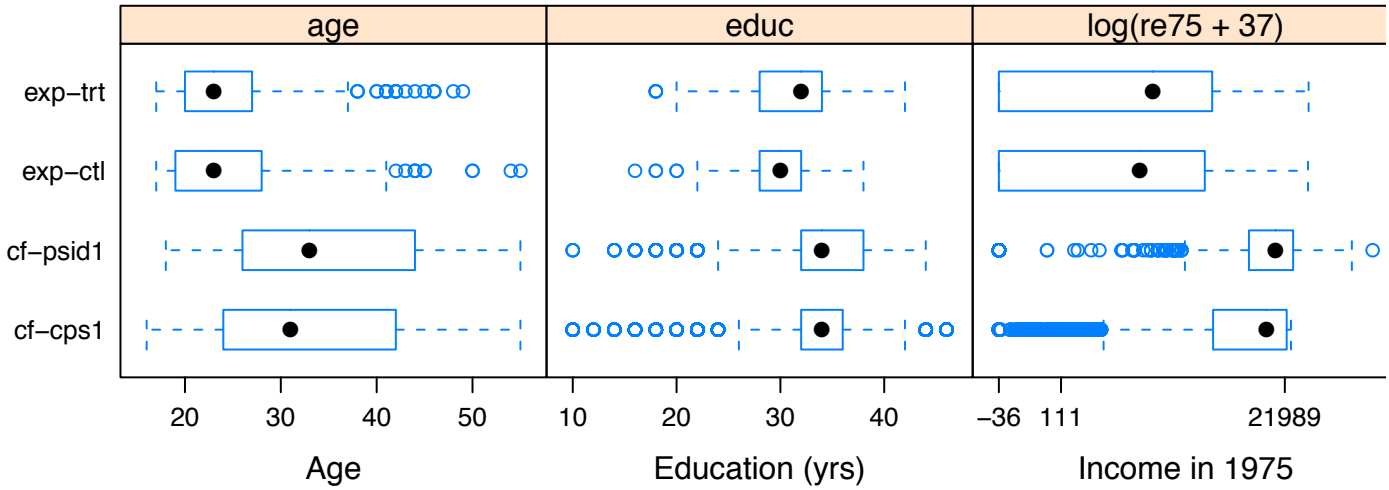

Figure 2: Boxplot comparisons, for Age, for Years of education, and for `log(re75+37)`. The logarithmic transformation leads to a more symmetric distribution. The plot for `log(re75+37` has a spike at zero.

Density plots do not work well with `educ`. Hence we try histograms.

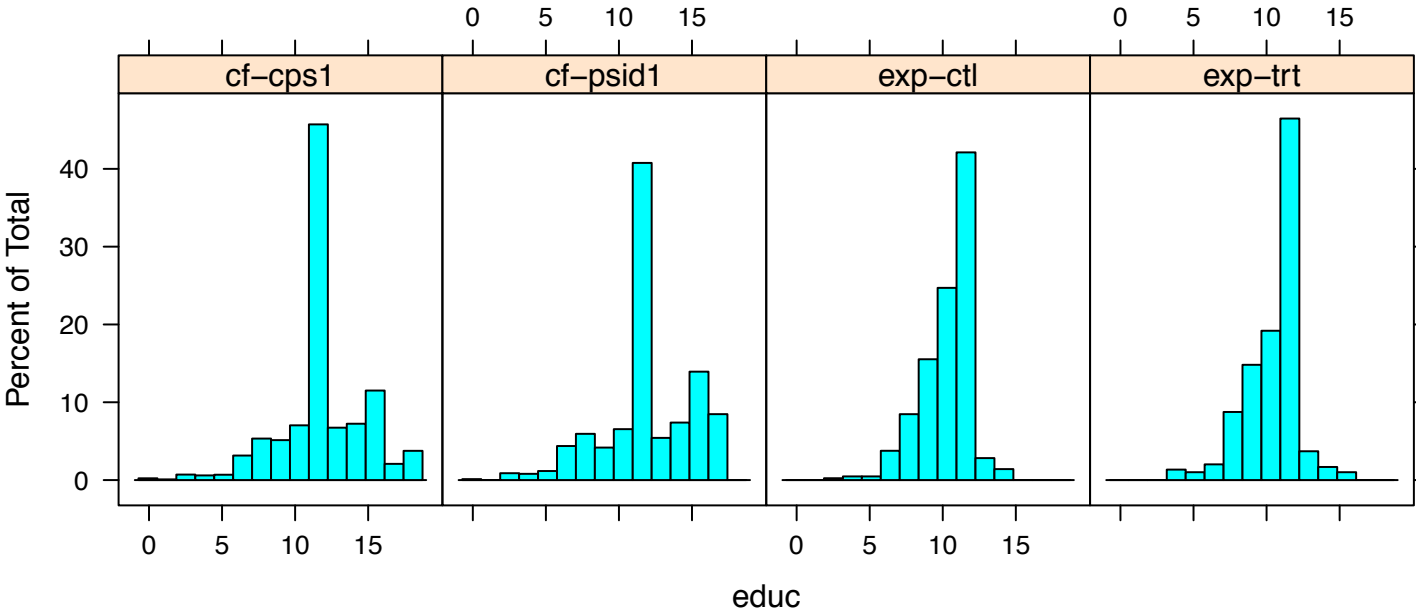

Figure 3: Histograms, used to compare distributions of Years of education.

- *With one density plot for all data area under the curve gives an indication the density of points, but the reader has to know or warned of the spike at zero, and to work on interpreting the graph. Alternatively, zeros can be omitted, with information provided on the proportion of zeros. Histograms are also acceptable (with a relatively large number of data points, their discreteness is not too much of an issue). They do not however readily lend themselves to overlaying.*
- *A logarithmic scale seems preferable for `re75`; otherwise the graph focuses unduly on a small number of very large values.*
- *Half the minimum non-zero value may be a reasonable offset when taking logarithms. Alternatively, choose the offset that leads to a near symmetric density plot.*

Note: The assignment did not ask for a graph for years of education, but it (or something like it) is needed for the comparison in Exercise 6.

5. Use tables to summarise differences in categorical variables between the two groups. [2 marks]

```
          black  hisp  marr nodeg
cf-cps1   0.074 0.072 0.712 0.296
cf-psid1  0.251 0.033 0.866 0.305
exp-ctl   0.800 0.113 0.158 0.814
exp-trt   0.801 0.094 0.168 0.731
```

6. What are the major differences between the four groups, as evident from examining individual columns? Comment especially on any differences in the pre-training variables, i.e., all except `re78`. [3 marks]
The proportions in all categorizations except ***hisp*** *are very different. In the experimental groups there are many more blacks, many more who had not graduated from high school, and many fewer married. Here is the comparison between the proportion who had fewer than 12 years education:*

```
  cf-cps1  cf-psid1   exp-ctl   exp-trt
0.2958354 0.3052209 0.8141176 0.7306397
```

Note also differences in the **age** *distribution and in the proportion earning no income in 1975 (those in the experimental groups were much younger, and more likely to have had no income in 1975. NB: You were asked to draw attention to major differences.*

7. Do any differences in the pre-training variables have implications for the way that you might analyse the data, or the reliance that you might put on the results? [2 marks]
The experimental control and treatment groups should, because of the randomization, be similar on all pre-treatment variables. The various comparisons above suggest that they are acceptably similar. There are however big differences between the experimental data and non-experimental controls. Unless there is some way to adjust for these differences, the non-experimental "controls" are clearly quite unsuitable for use as controls.

8. The aim of the study was to assess the effect of training. Is it best to base the comparison between treatment and control group i) on `re78` alone, or ii) on `re78 - re75`? Justify your answer. [2 marks]
One might expect some correlation between income in 1978 and income in 1975. The plot `plot(log(re78+37) ~ log(re75+37), data=nswdemo)` *suggests that there is a small correlation. Examination of the change in income might therefore give a fairer assessment. On the other hand, the contribution of a few large changes in income, from 1975 to 1978, may introduce undesirable noise. Some subtlety is required in deciding how best to proceed.*

[**TOTAL: 20 marks**]

Code

Setup

```
> opt <- options(SweaveHooks=list(fig=function(){par(cex.main=1.0, 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)}),
                            continue="  ")
> pdf.options(pointsize=8)
> library(DAAG)
```

Creation of a Suitable Data Objects

```
> ## Note the coding for the experimental control & treatment groups
> table(nswdemo$trt)
> ## Code the non-experimental controls as 2 (cps1) and 3 (psid1)
> cps1$trt <- rep(2, dim(cps1)[1])
> psid1$trt <- rep(3, dim(psid1)[1])
> allsets <- rbind(nswdemo, cps1, psid1)
```

```
> allsets$trt <- factor(c("exp-ctl", "exp-trt", "cf-cps1", "cf-psid1")
                        [allsets$trt+1])
> ## Check that the numbers in the different groups make sense
> table(allsets$trt)
```

Counts of number of missing values, or zeros

```
> ## Function that retrieves number of NAs for a single column
> countNAcol <- function(x)sum(is.na(x))
> ## Now use the function aggregate() to apply countNAcol() to
> ## allsets, indexed by trt
> aggregate(allsets, list(group = allsets$trt), FUN = countNAcol)

> countzeros <- function(x)sum(!is.na(x) & x==0)
> aggregate(allsets[, c("re74", "re75", "re78")], list(group=allsets$trt),
            FUN=countzeros)
```

Alternative

```
> ## Function that retrieves number of NAs for each column of a data frame
> countlist <- function(z, statsfun=length)sapply(z, statsfun)
>   # Notice that statsfun has been given a default argument
> ## Apply this to the allsets data frame
> sapply(split(allsets, allsets$trt), FUN=countlist, statsfun=countNAcol)
```

Alternative data object

```
> listsets <- list("cf-cps1"=cps1[,-1], "cf-psid1"=psid1[,-1],
                   "exp-ctl"=subset(nswdemo, trt==0)[,-1],
                   "exp-trt"=subset(nswdemo, trt==1)[,-1])
> ## Check that the numbers in the different groups make sense
> sapply(listsets, nrow)
> ## Now do the calculations with listsets
> sapply(listsets, FUN=countlist, statsfun=countNAcol)
```

Code for exercises

1. ...

2. (a) `> t(sapply(listsets, FUN=countlist, statsfun=countNAcol))`

   (b)
```
> num <- sapply(split(allsets[, c("re74","re75","re78")],
                      allsets$trt), FUN=countlist, statsfun=countzeros)
> pr <- sapply(split(allsets[, c("re74","re75","re78")],
                      allsets$trt), FUN=countlist,
             statsfun=function(x)countzeros(x)/length(na.omit(x)))
> cbind(num, round(pr,3))
```

> Note: The above uses a "put as much as possible onto one line" style of code that is compact and easy to manage, but which novices may find overly cryptic. The less condensed code that now follows is more explicit. It works with smaller units of data, which is a consideration if datasets are large relative to available memory. There is some redundancy, which should be removed if these calculations are repeated a large number of times. The more obvious ways to remove the redundancy lead to code that is, again, somewhat more cryptic.

```
> ## Longer, less cryptic and clearer code, for exptl data only
> namre <- c("re74","re75","re78")
> expctlz <- sapply(subset(nswdemo[, namre], nswdemo$trt==0), countzeros)
> exptrtz <- sapply(subset(nswdemo[, namre], nswdemo$trt==1), countzeros)
> numz <- cbind(expctl=expctlz, exptrt=exptrtz)
> expctlp <- sapply(subset(nswdemo[, namre], nswdemo$trt==0),
```

```
                        function(x)countzeros(x)/length(na.omit(x)))
> exptrtp <- sapply(subset(nswdemo[, namre], nswdemo$trt==1),
                        function(x)countzeros(x)/length(na.omit(x)))
> propz <- cbind("expctl-p"=expctlp, "exptrt-p"=exptrtp)
> cbind(numz, propz)

      expctl exptrt  expctl-p  exptrt-p
re74    195    131 0.7500000 0.7081081
re75    178    111 0.4188235 0.3737374
re78    129     67 0.3035294 0.2255892
```

3. (a)
```
> round(with(nswdemo, sapply(split(re74,trt),
                          function(x)countNAcol(x)/length(x))), 3)
```

   (b)
```
> round(sapply(split(allsets[, c("re74","re75","re78")],
                  allsets$trt), FUN=countlist,
             statsfun=function(x)countzeros(x)/length(x)), 3)
```

   (c)
```
> varnam <- c("age","educ","re74","re75","re78")
> ran <- sapply(split(allsets[, varnam],
                  allsets$trt), FUN=countlist,
                  statsfun=function(x)range(x,na.rm=TRUE))
> rownames(ran) <- paste(rep(varnam, rep(2,length(varnam))),"-",
                           rep(c("l","u"), length(varnam)), sep="")
> print(round(t(ran),0))
```

4.
```
> library(lattice)
> at1 <- pretty(allsets$age,5)
> labs1 <- paste(at1)
> at2 <- pretty(allsets$educ,5)
> labs2 <- paste(at1)
> at3 <- with(allsets, pretty(log(re75+37), 3))
> at3 <- c(log(37), at2[at2>log(75)])
> labs3 <- paste(round(exp(at2) - 37, 0))
> print(densityplot(~age+educ+log(re75+37), groups=trt, auto.key=list(columns=4),
               scales=list(x=list(at=list(at1,at2,at3),
                           labels=list(labs1,labs2,labs3)), relation="free"),
                    xlab=c("Age", "Education (yrs)", "Income in 1975"),
                    data=allsets))

> print(bwplot(trt~age+educ+log(re75+37), outer=TRUE,
               scales=list(x=list(at=list(at1,at2,at3),
                           labels=list(labs1,labs2,labs3), relation="free")),
               xlab=c("Age", "Education (yrs)", "Income in 1975"),
               data=allsets))

> print(histogram(~educ | trt, data=allsets, layout=c(4,1)))
```

5.
```
> cat2prop <- function(x) if(is.factor(x))sum(x==levels(x)[2])/length(x) else
                    sum(x==max(x))/length(x)
> contnum <- match(c("trt",varnam), names(allsets))
> t(round(sapply(split(allsets[, -contnum],
               allsets$trt), FUN=countlist, statsfun=cat2prop), 3))

> ## Proportion with less than 12 years education
> with(allsets, sapply(split(educ,trt), function(x)sum(x<12)/length(x)))
```

Use of the *doBy* Package

The `summaryBy()` function has extensive abilities for calculating summary information. It returns a data frame. The following does the calculation required to calculate, for each of the nominated splits of the data and for each of the nominated columns, the number of zeros:

```
> ## The "0s" gets added to the names of the variables
> countzeros <- function(x)c("0s"=sum(!is.na(x) & x==0))
> num0 <- summaryBy(re74 + re75 + re78 ~ trt, data=allsets, FUN=countzeros)
```

The first column is a factor, with one row for each level of `trt`. The level names are better used to label the rows. The first column can then be omitted, leaving a data frame that has all columns numeric.

```
> row.names(num0) <- as.character(num0[,1])
```

Then `print(num0[, -1])` prints with the levels of `trt` used as row names, while `print(t(num0[, -1]))` prints without the quotes that would otherwise appear.
[The object `t(num0)` is a matrix of character. It has to be character because its first row is character. The command `print(t(num0), quote=FALSE))` can be used to print it without the quotes.]

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}
```

Setting options globally: The following is an example.

```
<<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)
@%
```

This reduces the margins around graphs (`mar=c(4.1,3.6,1.6,1.1)` created using base graphics, gives plots that are square (`pty="s"`), moves the axis labels and tick labels in somewhat (`mgp=c(2.25,0.5,0)`), and reduces the length of tick marks (`tck=-0.025`). The call to `pdf.options()` sets the pointsize for postscript and pdf graphs. Note that these settings have no effect for *lattice* and *ggplot2* graphs.

Also possible are `options(prompt=" ", continue=" ")`, or `options(continue=" ")` Use the first of these options only once everything is working. Also, include near the end of the document:

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

To restore the prompt and continuation prompt from the command line, perhaps because the document has not processed properly through Sweave, type:

```
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 use of `@ %` rather than `@` is optional. The `%` tells editors to respect the line feed when reformatting.