Dynamic Documents with knitr

Iowa State University

April 9, 2014

knit what?

- Adds (knits) R code into your document so that all of the files for a particular experiment are in one place
- Compiling the document also runs the R code, so one button press runs your statistical analysis and places it into your document
- Integrates R output with the document text so that pictures, tables, and numeric data are all dependent on the R code
- Produces beautiful, seamless integration of statistical analysis with scientific writing

Reproduceable Research

- Journals often require code and data to be included in a paper submission so the analysis can be reproduced
- Statistical analyses are often re-used for multiple similar experiments
- We don't always comment code when we write it - knitr provides context for the code by including the writeup with the code

Setup

- LaTeX: If you don't have this installed, you will need to use our computers.
- knitr: use `install.packages("knitr")` to install the knitr package in R
- RStudio:

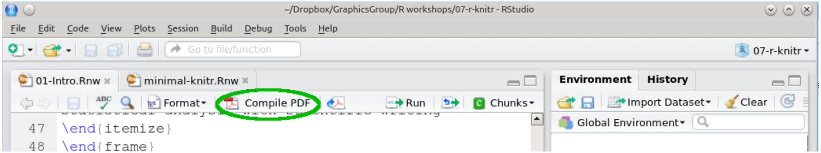
  Tools –> Global Options –> Sweave –> Weave Rnw files using knitr

  (This tells RStudio to use knitr to compile the document)

Example

Open minimal-knitr.Rnw in RStudio and compile it

Hint: use the "Compile PDF" button at the top of the text window in RStudio



```
\end{itemize}
\end{frame}
```

R Code "Chunks"

knitr knows that text is R code if it is included in a "chunk":

```
«chunk-label, options=...»=

     your code goes here

@
```

Your Turn

Change minimal-knitr.Rnw so that the second code chunk (named 'boring-random') generates 30 random numbers and computes their sum.

Make sure the code chunk compiles!

Adding graphs to documents with knitr

```
\centering
<<chunk-label, options=...>>=

      make a plot here

@
```

Your Turn

Add a new code chunk to minimal-knitr.Rnw that generates a sample of 50 observations from a normal distribution and plot a histogram.

```
x <- rnorm(50)

library(ggplot2)
qplot(x, geom="histogram")
```

Make sure the code chunk compiles!

knitr code chunk options

We may want to remove the messages (the italic text) from our R output in the pdf. Or, we might not want to show the code at all.

knitr has options to give you greater control over what is shown in the compiled document.

Let's look at these options in the documentation.

knitr code chunk options

- `eval=FALSE`: don't run the code
- `echo=FALSE`: don't show the code

- `warning=FALSE`: don't add R's warnings to the pdf
- `message=FALSE`: don't show messages from R in the pdf

- `include=FALSE`: don't include any R output in the document
  Pictures are still saved and can be included with `\includegraphics{}`.
- `tidy=FALSE`: don't tidy code or change spacing
- `cache=FALSE`: re-run all of the code every time.

Your Turn

Explore some of the options in knitr code chunks.

- Turn off messages and warnings for the histogram we generated last time
- Experiment with `tidy=TRUE` - what changes occur?
- Can you make knitr show only the plot, without code, messages, errors, etc.?

Common Options for Pictures

- `fig.width=7, fig.height=7`: dimensions of the saved image
- `out.width=‘\\textwidth’, out.height=‘300px’`: dimensions of the image in the LaTeX document.
  Note: \\must be used to generate a single slash in LaTeX.

- `fig.path="path/to/save/figures"`: Change where figures are stored
- `dpi=72`: resolution of the saved image

- `fig.keep=‘‘all"`: Which figures to keep
  Options: ‘high’, ‘none’, ‘all’, ‘first’, ‘last’
- `fig.show=‘‘asis’’`: how to arrange plots
  Options: ‘asis’, ‘hold’, ‘animate’, ‘hide’

Your Turn

- Generate both a histogram and a density plot in the same code chunk
  try different `fig.keep` and `fig.show` options
- Change the size of both plots so they will fit on the same page
  Hint: use `out.width=“.48\\textwidth”`
- Use `include=FALSE` and include the figure in LaTeX directly:

```
\begin{figure}[h]\centering
\includegraphics[width=.5\linewidth]
    {figure/codechunkname}
\end{figure}
```

More Common Options for Pictures

- `fig.cap="caption text"`: Text for figure caption
- `fig.align="center"`: Alignment of the figure on the document page.
  Options 'left', 'right', 'center'
- `fig.pos="htbp"`: Latex positioning options - "h" for here, "t" for top, "b" for bottom, "p" a page of floats.
- `fig.env=`: LaTeX environment for figures (i.e. SCfigure, marginfigure)
- `fig.lp="fig:"`: Label prefix for figure label: the actual label will concatenate this prefix and the chunk label. So to reference a figure in chunk "plot" in a LaTeX paragraph, you would use `\ref{fig:plot}` to get the plot number.

Setting commonly used options for all chunks

It is common to create a setup chunk at the beginning of a LaTeX document (i.e. before the \begin{document} line) to load packages and set default options.

```
<<setup,cache=FALSE,echo=FALSE>>=

opts_chunk$set(fig.width=6, fig.height=6, tidy=FALSE, echo=

@
```

Your Turn

Try setting default options at the beginning of our Rnw file, and see how things change.

Notice that you can always override default options within each chunk header.