

Lecture 1 focuses on reproducible documentation and dynamic report generation. Material is taught using the **Knitr** package and examples include writing RMarkdown and Sweave files. This handout provides helpful reference material for creating dynamic HTML and L^AT_EX reports by embedding R code into RMarkdown and Sweave files using Knitr options and functionality.

Embed R code

Insert an R code chunk in an R Markdown file using this syntax:

```
```{r my-chunk-label}  
y <- 1 + 1
print(y)
```
```

Insert an R code chunk in an R Sweave file using this syntax:

```
<<my-chunk-label>>=  
y <- 1 + 1  
print(y)  
@
```

All chunk labels must be unique. Unnamed chunks will be assigned a unique label.

Chunk options

Below is a selection of Knitr chunk options useful for the in-class exercises and beyond. For the exhaustive list of chunk options visit Yihui's Knitr site, options page: <http://yihui.name/knitr/options>

Remember these are knitr options, so they work regardless of whether you are creating a LaTeX or HTML document type. The options are shown with their default values and option argument types are provided in parentheses. Option values must be valid R expressions, as shown below. When including chunk options in your .Rnw or .Rmd files, they must be written on a single line; no line breaks allowed. Avoid spaces and periods "." in chunk labels.

eval=TRUE: (logical) whether to evaluate the code chunk; it can also be a numeric vector to select which R expression(s) to evaluate, e.g. **eval=c(2,3)** or **eval=-(4:6)**

cache=FALSE: (logical) whether to cache a code chunk; good for time-consuming routines; cached code chunks are evaluated the first time and their output saved to a database file and accessed by dependent chunks, then this chunk is skipped all subsequent times the report file is compiled; **cache=TRUE** does exactly what you want it to in that if you later make a change to the cached chunk, it will re-evaluate the cached chunk, store the results and then skip over the R code in the cached chunk in all subsequent knitting.

Chunk option example using **eval**: This example is written in Sweave (.Rnw) syntax, but the chunk option location is the same for a .Rmd. We use **eval=FALSE** to turn evaluation of the R code off in this chunk, but source code will still be included in the final document.

```
<<storeY, eval=FALSE>>=  
y <- 1+1  
@
```

Text results

echo=TRUE: (logical or numeric) whether to include R source code in the output file; use a logical value to turn on/off the source code or provide a numeric vector that selects which R expression(s) to echo in a code chunk

include=TRUE: (logical) whether to include the chunk output in the final output document; if **include=FALSE** nothing will be written to the output file, but code is evaluated and plots files are generated so you can manually insert figures

results='markup': the default setting; mark up the results using a LaTeX environment; similar to **verbatim** in Sweave

results='asis': write raw results (as is) from R into the output document

results='hold': hold all of the R output pieces and push them to the end of a chunk

`results='hide'`: hide normal R output (not warnings, messages, or errors)
`error=TRUE`: (logical) whether to preserve errors; by default evaluation will not stop in the case of errors; set option to `FALSE` to cause R to stop on errors; similar options are `message=TRUE` and `warning=TRUE`

Code decoration

`comment='##'`: (character) the prefix knitr places in front of source code output for easy reading and copy/paste; set `comment=NA` to disable this feature
`prompt=FALSE`: (logical) whether to include the prompt characters in the R code
`highlight=TRUE`: (logical) whether to highlight the source code

Figure options

`fig.keep='high'`: (character) how plots in a chunk should be kept; the default option is to only keep high-level plots as output to the report by merging all low-level changes to a single plot into one high-level plot; contrast the default value to `'all'` which will keep all plots including `'iterations'` of the same plot; other argument values: `none`, `first last`, see Yihui's site for more information

`fig.show='asis'`: (character) how to show the plots; the default will show them immediately after the code within a chunk, exactly where they were generated; contrast the default argument to `'hold'` which will hold all plots generated within that chunk until the end; use `'hide'` to generate plots but hide them from the output file; use `'animate'` to wrap all plots in a chunk into an animation

`fig.width/fig.height=7`: (numeric) dimensions can be specified in inches (shown) or cm: `fig.width/fig.height='8cm'`, to be used in graphics device

`out.width/out.height=NULL`: (NULL or character) set width/height of the plot in the final output file; can be different from its real `fig.width/fig.height` to scale the plots in the final output file; in LaTeX supply a size in inches or cm, in HTML supply a pixel size `300px`; I encourage you to play around with setting different values of `fig.width/height` and `out.width/height` in a chunk to better understand these options

`fig.cap=NULL`: (NULL or character) write caption text

`fig.align='default'`: (character) alignment of figures in the output document; other possible values are `left`, `right`, `center`

`dev='pdf'` for LaTeX, `dev='png'` for HTML/markdown: (character) the function name to be used as a graphical device to record plots; includes all graphics devices in base R as well as those in Cairo and many more, see Yihui's knitr site for full description

`interval=1`: (numeric) number of seconds to pause between animation frames; see documentation of the `animate` package:

<http://www.ctan.org/tex-archive/macros/latex/contrib/animate>

Global chunk options

To set global chunk options use `opts_chunk$set()`.

For example including the following code early in your Sweave document

```
<<setGlobalOptions, include=FALSE>>=
```

```
opts_chunk$set(comment=NA)
```

```
@
```

will cause all following code chunks to omit the prefix comment symbols from chunk results. Global chunk options can be overridden for any chunk by specifying different values inside that chunk.

Plots

R code remains the same inside of a chunk when you want to include a plot:

```
```{r scatterplot}
x <- rnorm(100)
y <- 2*x + rnorm(100)
plot(x,y)
```
```

Use figure options (above) for fine plotting control.

The example above is for .Rmd. The functionality is the same in a .Rnw file, you just need to use the correct R chunk syntax: (`<<>= @`) to declare “here lies R code”. See R code section above.

When plots are generated using knitr, a folder named **Figure** is created in the source file directory. This folder contains all of the plots generated in that knitr file in the default or specified format using chunk option **dev**, more information above. I encourage you to look at this file now.

Tables in an HTML report

Use **pander** to export tables to HTML.

```
```{r do_table, results='asis'}
library(pander)
panderOptions("table.style", "rmarkdown")
mydf <- data.frame(id = 1:10, var1 = rnorm(10), var2 = runif(10))
pander(mydf)
```
```

Pander works very well with a Markdown workflow, and marks up vectors (example below), matrices, etc. Read the **pander** documentation for more information.

```
```{r markupVector, results='asis'}
library(pander)
pander(1:10)
```
```

You can also use **xtable** to export tables to HTML. To use **xtable** in RMarkdown you need to specify **type = 'html'**; the default is **type = 'latex'**.

```
```{r do_table, results='asis'}
library(xtable)
mydf <- data.frame(id = 1:10, var1 = rnorm(10), var2 = runif(10))
print(xtable(mydf), type = "html")
```
```

Tables in a \LaTeX report

Use **xtable** to export tables to LaTeX, this is your best option.

```
<<do_table, results = "asis">>=
library(xtable)
mydf <- data.frame(id = 1:10, var1 = rnorm(10), var2 = runif(10))
print(xtable(mydf))
@
```

Exercises

Exercise 1: Use Knitr to create a dynamic web-based report (HTML)

Open the RMarkSolution.html, it should automatically open in a new tab in your web browser. Please recreate this report exactly using the stub R Markdown file provided named RMarkExercise.Rmd. Use the lecture slides, this handout, or the World Wide Web.

Exercise 2: Use Knitr to create a dynamic LaTeX report

Open the LaTeXSolution.pdf and study it. Please recreate this report exactly using the stub .Rnw file provided named LaTeXSolution.Rnw. Use the lecture slides, this handout, or the World Wide Web.

Convert your old Sweave files to knitr-compatible documents

In case you want to update analysis or results in your old Sweave files using knitr, you can easily convert them into knitr-compatible .Rnw files using the R command `Sweave2knitr()` which is part of the knitr package. To use, set your working directory to the oldFile.Rnw location, make sure the knitr library is loaded, then run `Sweave2knitr("oldFile.Rnw")` in the R console. In the same directory you should see a new .Rnw with name oldFile-knitr.Rnw. Knitr also writes all the changes to the R console so you can see/learn how to line-by-line convert to knitr-compatible files.

Markdown quick reference

If you have a .Rmd file open in R studio, hit the **MD** button for a quick reference guide, or search for your favorite one on the internet (there are many).

Definitions to remember

Knitr is an engine for dynamic report generation that enables the integration of R code into LaTeX, HTML, Markdown, AsciiDoc, and reStructuredText documents.

Markdown is a text-to-HTML conversion tool for web-writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML or HTML. Thus, Markdown is two things: (1) a plain text formatting syntax; and (2) a software tool, written in Perl, that converts plain text formatting to HTML.

R Markdown (a variant of Markdown) is a format that allows embedded R code/output/plots in web-based reports.

LaTeX is a high-quality typesetting system. Think of every statistics text book you know: probably created using LaTeX.

Pandoc is a universal document converter.

Useful websites

A lite coding style guide from Hadley Wickham: <http://adv-r.had.co.nz/Style.html>

Helpful list of all R packages that facilitate reproducible research:

<http://cran.r-project.org/web/views/ReproducibleResearch.html>

A good start to learn about R Markdown:

http://kbroman.github.io/knitr_knutshell/pages/Rmarkdown.html

Best site on Knitr written by the package author: <http://yihui.name/knitr/>

A good start to learn about Markdown: <http://daringfireball.net/projects/markdown/basics>

Excellent resource for TeX: <http://en.wikibooks.org/wiki/LaTeX>

is an online wiki book has everything you need to know about LaTeX. Similar to R, it is very easy to search for LaTeX help in places like StackOverflow or StackExchange.

Pandoc website: <http://johnmacfarlane.net/pandoc/>

Optional lecture extensions

1. Use Markdown in email correspondence: go to <http://markdown-here.com/> and download the package for your browser.

2. Publish an HTML file to the web using Rpubs: <http://markdown-here.com/>