

# Knitr?? What about Sweave?

December 11-12, 2013.

# Knitr - the new Sweave

- Sweave original literate programming engine for R (or S) written by Friedrich Leisch

# Knitr - the new Sweave

- Sweave original literate programming engine for R (or S) written by Friedrich Leisch
  - ▶ still works perfectly well

# Knitr - the new Sweave

- Sweave original literate programming engine for R (or S) written by Friedrich Leisch
  - ▶ still works perfectly well
- knitr re-implementation of sweave written by Yihui Xie

# Knitr - the new Sweave

- Sweave original literate programming engine for R (or S) written by Friedrich Leisch
  - ▶ still works perfectly well
- knitr re-implementation of sweave written by Yihui Xie
  - ▶ homepage: <http://yihui.name/knitr/>

# Knitr - the new Sweave

- Sweave original literate programming engine for R (or S) written by Friedrich Leisch
  - ▶ still works perfectly well
- knitr re-implementation of sweave written by Yihui Xie
  - ▶ homepage: <http://yihui.name/knitr/>
- minor differences in \*.rnw files

# Knitr - the new Sweave

- Sweave original literate programming engine for R (or S) written by Friedrich Leisch
  - ▶ still works perfectly well
- knitr re-implementation of sweave written by Yihui Xie
  - ▶ homepage: <http://yihui.name/knitr/>
- minor differences in \*.rnw files
- knitr much more flexible

# Knitr - the new Sweave

- Sweave original literate programming engine for R (or S) written by Friedrich Leisch
  - ▶ still works perfectly well
- knitr re-implementation of sweave written by Yihui Xie
  - ▶ homepage: <http://yihui.name/knitr/>
- minor differences in \*.rnw files
- knitr much more flexible
  - ▶ much better error handling



# Knitr - the new Sweave

- Sweave original literate programming engine for R (or S) written by Friedrich Leisch
  - ▶ still works perfectly well
- knitr re-implementation of sweave written by Yihui Xie
  - ▶ homepage: <http://yihui.name/knitr/>
- minor differences in \*.rnw files
- knitr much more flexible
  - ▶ much better error handling
  - ▶ markdown and html

# Knitr - the new Sweave

- Sweave original literate programming engine for R (or S) written by Friedrich Leisch
  - ▶ still works perfectly well
- knitr re-implementation of sweave written by Yihui Xie
  - ▶ homepage: <http://yihui.name/knitr/>
- minor differences in \*.rnw files
- knitr much more flexible
  - ▶ much better error handling
  - ▶ markdown and html
  - ▶ extensible to other languages

# Knitr - the new Sweave

- Sweave original literate programming engine for R (or S) written by Friedrich Leisch
  - ▶ still works perfectly well
- knitr re-implementation of sweave written by Yihui Xie
  - ▶ homepage: <http://yihui.name/knitr/>
- minor differences in \*.rnw files
- knitr much more flexible
  - ▶ much better error handling
  - ▶ markdown and html
  - ▶ extensible to other languages
- all subsequent examples will refer to knitr

# Differences between knitr and Sweave

- no \*.sty file to register with knitr

# Differences between knitr and Sweave

- no \*.sty file to register with knitr
- knitr chunk options are R-expressions

# Differences between knitr and Sweave

- no \*.sty file to register with knitr
- knitr chunk options are R-expressions
- output code automatically 'pretty printed'

# Differences between knitr and Sweave

- no \*.sty file to register with knitr
- knitr chunk options are R-expressions
- output code automatically 'pretty printed'
- errors and warnings are rendered directly in document

# Differences between knitr and Sweave

- no \*.sty file to register with knitr
- knitr chunk options are R-expressions
- output code automatically 'pretty printed'
- errors and warnings are rendered directly in document
- graphics



# Differences between knitr and Sweave

- no \*.sty file to register with knitr
- knitr chunk options are R-expressions
- output code automatically 'pretty printed'
- errors and warnings are rendered directly in document
- graphics
  - ▶ grid based graphics render without `print()`

# Differences between knitr and Sweave

- no \*.sty file to register with knitr
- knitr chunk options are R-expressions
- output code automatically 'pretty printed'
- errors and warnings are rendered directly in document
- graphics
  - ▶ grid based graphics render without `print()`
  - ▶ multiple graphics per chunk

# Differences between knitr and Sweave

- no \*.sty file to register with knitr
- knitr chunk options are R-expressions
- output code automatically 'pretty printed'
- errors and warnings are rendered directly in document
- graphics
  - ▶ grid based graphics render without `print()`
  - ▶ multiple graphics per chunk
  - ▶ fine grained control of figure environment

# Literate Programming

*Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to humans what we want the computer to do.*

– Donald E. Knuth, *Literate Programming*, 1984

# Basics

- both knitr and sweave realize literate programming by embedding analysis code directly in source code of final document
- \*.rnw - 'R-noweb' file
  - ▶ contain both  $\text{\LaTeX}$  and embedded R code
- code introduced in 'chunks' or using `\Sexpr{}`
- same basic principles apply to other formats and documents types

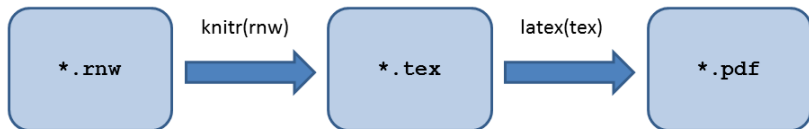
# rnw file format and emacs

- ESS supports rnw files - major mode changes depending on location point
  - ▶ latex mode outside of chunks
  - ▶ ESS[S] mode inside of chunks

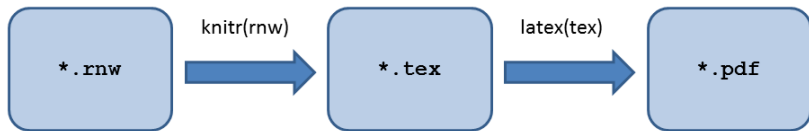
## Compiling to pdf

M-r      run knitr current buffer (M-s to sweave)  
M-S-p    compile tex file pdf

# Compiling to pdf



# Compiling to pdf





# Chunks

- blocks of R code embedded directly in  $\text{\LaTeX}$  file
- delineated by `<<>>= ... @`

## Example chunk:

```
% latex code
<<>>=
wfddata <- read.csv('whitefish.csv', header=TRUE)
ages <- unique(wfddata$AGE)
@
% latex code
```

# Chunk Options

- chunk options control how sweave/knitr process code and handle results
- of the form:
  - ▶ `<<label, option1=foo, option2=bar>>=`
- knitr has many more chunk options than sweave
  - ▶ almost too many
  - ▶ many overlap with  $\text{\LaTeX}$
- must be on one line (no line breaks)
- `opts_chunk$set()` can be used to set default value for all subsequent chunks in a document

# Common Chunk Options

- `label`

# Common Chunk Options

- `label`
  - ▶ first chunk option

# Common Chunk Options

- `label`
  - ▶ first chunk option
  - ▶ optional, but highly recommended

# Common Chunk Options

- `label`
  - ▶ first chunk option
  - ▶ optional, but highly recommended
- `eval`

# Common Chunk Options

- `label`
  - ▶ first chunk option
  - ▶ optional, but highly recommended
- `eval`
  - ▶ TRUE/FALSE OR numeric vector

# Common Chunk Options

- `label`
  - ▶ first chunk option
  - ▶ optional, but highly recommended
- `eval`
  - ▶ TRUE/FALSE OR numeric vector
  - ▶ controls whether code chunk is processed by R interpreter



# Common Chunk Options

- `label`
  - ▶ first chunk option
  - ▶ optional, but highly recommended
- `eval`
  - ▶ TRUE/FALSE OR numeric vector
  - ▶ controls whether code chunk is processed by R interpreter
- `echo`

# Common Chunk Options

- `label`
  - ▶ first chunk option
  - ▶ optional, but highly recommended
- `eval`
  - ▶ TRUE/FALSE OR numeric vector
  - ▶ controls whether code chunk is processed by R interpreter
- `echo`
  - ▶ TRUE/FALSE OR numeric vector

# Common Chunk Options

- `label`
  - ▶ first chunk option
  - ▶ optional, but highly recommended
- `eval`
  - ▶ TRUE/FALSE OR numeric vector
  - ▶ controls whether code chunk is processed by R interpreter
- `echo`
  - ▶ TRUE/FALSE OR numeric vector
  - ▶ should expression in code chunk appear in rendered document

# Common Chunk Options (cont'd)

- `results`
  - ▶ one of 'asis', 'hide', 'markup', 'hold'
  - ▶ should results of code chunk appear in rendered document
- `include`
  - ▶ TRUE/FALSE
  - ▶ if `include==FALSE` nothing from chunk will be written into document
  - ▶ code in chunk is evaluated, not included
  - ▶ ideal for figures and analysis code
- `error, warning, message`
  - ▶ TRUE/FALSE
  - ▶ whether to include or suppress errors, warning or messages in final document

# Chunk examples:

Performs an action, no  $\text{\LaTeX}$  returned:

```
<<label=fit_model, include=FALSE>>=  
...  
@
```

Include  $\text{\LaTeX}$  results from R:

```
<<label=meanTable, results='asis', echo=FALSE>>=  
...  
@
```

# `Sexpr{}`

- used to insert values inline (mid-sentence)
- cannot contain line breaks
- longer expressions should be defined in preceding chunks

## An Example:

The value of `\pi = \Sexpr{pi}`.

will render as:

The value of  $\pi = 3.1416$ .

# Tables

- number of packages that render R objects as latex tables

# Tables

- number of packages that render R objects as latex tables
  - ▶ xtable (Dahl, 2013)



# Tables

- number of packages that render R objects as latex tables
  - ▶ xtable (Dahl, 2013)
  - ▶ Hmisc (Harrel, 2013)

# Tables

- number of packages that render R objects as latex tables
  - ▶ xtable (Dahl, 2013)
  - ▶ Hmisc (Harrel, 2013)
  - ▶ tables (Murdoch, 2013)

# Tables

- number of packages that render R objects as latex tables
  - ▶ xtable (Dahl, 2013)
  - ▶ Hmisc (Harrel, 2013)
  - ▶ tables (Murdoch, 2013)
- all will render data frames as  $\text{\LaTeX}$  tables

# Tables

- number of packages that render R objects as latex tables
  - ▶ xtable (Dahl, 2013)
  - ▶ Hmisc (Harrel, 2013)
  - ▶ tables (Murdoch, 2013)
- all will render data frames as  $\text{\LaTeX}$  tables
- large number of arguments for flexibility

# Tables

- number of packages that render R objects as latex tables
  - ▶ xtable (Dahl, 2013)
  - ▶ Hmisc (Harrel, 2013)
  - ▶ tables (Murdoch, 2013)
- all will render data frames as  $\text{\LaTeX}$  tables
- large number of arguments for flexibility
- support for other R-object differs by package

# Tables

- number of packages that render R objects as latex tables
  - ▶ xtable (Dahl, 2013)
  - ▶ Hmisc (Harrel, 2013)
  - ▶ tables (Murdoch, 2013)
- all will render data frames as  $\text{\LaTeX}$  tables
- large number of arguments for flexibility
- support for other R-object differs by package
- admb objects - require custom function to convert to data frame or write latex directly

# Tables (cont'd)

## Linear Model:

```
<<label=M2, results='asis'>>=
  cap.txt = "Coefficient estimates, standard errors
            and associated p-values for Model 2."
  xtable(M2, caption = cap.txt, label="tbl:M2")
@
```

## Custom Function:

```
%Mortality Summary
<<label=Morttable, results='asis'>>=
Mort.summary(scaa, latex=TRUE)
@
```

# Figures

- two different approaches:
  - ▶ chunk option `include=TRUE`
  - ▶ explicitly create figure in R and figure environment in  $\text{\LaTeX}$
- handling figures much improved in knitr
  - ▶ still too much magic



# Figures

- two different approaches:
  - ▶ chunk option `include=TRUE`
  - ▶ explicitly create figure in R and figure environment in  $\text{\LaTeX}$
- handling figures much improved in knitr
  - ▶ still too much magic

*“Explicit is better than implicit.”*  
– Tim Peters, *Zen of Python*

# Figures - graphicspath

- $\text{\LaTeX}$  variable `\graphicspath{}` can be set in the preamble and controls where  $\text{\LaTeX}$  looks for figures
- recommend setting `\graphicspath{}` in first chunk

## `graphicspath{}`

```
<<label=setup, include=FALSE>>=
library(knitr)
# set global chunk options
fig.path <- 'figures/'
opts_chunk$set(fig.path=fig.path, fig.align='center')
@

\graphicspath{{\Sexpr{fig.path}}}
```

# Figures - example chunks

## R-chunk

```
<<label=plot_fit, include=FALSE>>=  
pdf("LengthAtAge.pdf", height=5, width=5)  
plot(wfdata$AGE, wfdata$FLEN, xlab="Age", ylab="Length")  
lines(predicted$AGE, predicted$FLEN, col='blue')  
dev.off()  
@
```

## L<sup>A</sup>T<sub>E</sub>X

```
\begin{figure}  
  \begin{center}  
    \includegraphics[width=\textwidth]{LengthAtAge}  
  \end{center}  
  \caption{Mean size at age ...}  
  \label{fig:plot_fit}  
\end{figure}
```

# Figure and Table Captions

- captions in scientific writing are often verbose
  - ▶ numerous embedded values

## One way:

```
A <- pi
B <- sqrt(pi)
lake <- 'Lake Huron'
fyear <- 1990
lyear <- 2003

caption <- paste("This is a long figure caption where A = ",
                 round(A,3), " and B = ", round(B,5),
                 " samples where collected from ", lake,
                 " between ", fyear, " and ", lyear, sep="")
```

# Figure and Table Captions (cont't)

## A another way:

```
A <- pi
B <- sqrt(pi)
lake <- 'Lake Huron'
fyear <- 1990
lyear <- 2003

caption <- "This is a long figure caption where A = %.3f and
           B = %.5f. Samples where collected from %s between
           %i and %i."

caption <- sprintf(caption, A, B, lake, fyear, lyear)
```

# knitr template

- yasnippet template has been provided in workshop configuration
- attempts to provide sensible default styles and options:
  - ▶ sets figure directory in both  $\text{\LaTeX}$  and R
  - ▶ helper functions
  - ▶ running headers and footers
  - ▶ cjfas bibliography style
- in emacs create a new \*.rnw file and type knitr<TAB>

# Is it worth it?

- depends on:

# Is it worth it?

- depends on:
  - ▶ number of reports



# Is it worth it?

- depends on:
  - ▶ number of reports
  - ▶ number of times they need to be recreated or updated

# Is it worth it?

- depends on:
  - ▶ number of reports
  - ▶ number of times they need to be recreated or updated
  - ▶ report complexity

# Is it worth it?

- depends on:
  - ▶ number of reports
  - ▶ number of times they need to be recreated or updated
  - ▶ report complexity
- personal choice

# Is it worth it?

- depends on:
  - ▶ number of reports
  - ▶ number of times they need to be recreated or updated
  - ▶ report complexity
- personal choice
- definite long-term savings

# Is it worth it?

- depends on:
  - ▶ number of reports
  - ▶ number of times they need to be recreated or updated
  - ▶ report complexity
- personal choice
- definite long-term savings
  - ▶ require short term investment

# Recap

- knitr is the new Sweave
- sweave and knitr are literate programming functions for R
- R code is embedded directly in  $\text{\LaTeX}$  source documents
  - ▶ in line using `\Sexpr{}`
  - ▶ or in chunks
- chunks options determine how the code and its results are handled
- tables can be created using `Hmisc`, `xtable` or `tables` R-packages
- figures can be handled directly by knitr or explicitly with both R and  $\text{\LaTeX}$  code blocks