1 Syntax

format       start                          end                 inline            output
---          -------                          -------             -------            -------
Rnw          <<<*>>=                           @                  \$expr(x)         \TeX
Rmd          `{r *}`                          MD                 `r x`             
Rhtml        `<!-begin.rcode --> end.rcode--->` HTML<br>`!-rinline x--->` HTML
Rst          `.. {r *}`                       reST               `r x`             
Rtex          `% begin.rcode --> end.rcode`       \TeX<br>`\linenx(x)` \TeX
RasciiDoc    // begin.rcode // end.rcode       AsciiDoc<br>`+rx+` AsciiDoc<br>`+r x+` AsciiDoc
Rtex          ### begin.rcode ### end.rcode     Textile<br>`@r x@` Textile<br>`% x %` Textile
brew         # begin.rcode # end.rcode         text<br>`% x %` text<br>`% x %` text

* denotes local chunk options, e.g. `<label, eval=FALSE>>=; x denotes inline R code, e.g. `r 1+2` (MD stands for Markdown)

2 Minimal Examples

2.1 Sweave (*.Rnw)

\documentclass{article}
\begin{document}

Below is a code chunk.

```
<<foo, echo=TRUE>>=
z = 1+1
plot(cars)
```

The value of z is \$z\$.  \$\text{The value of z is } z\$.

\end{document}

2.2 R Markdown (*.Rmd)

Hi _markdown_!

```
```
\`r foo, echo=TRUE\`
```

\text{The value of z is } z\$.

2.3 Brew (*.brew)

The value of pi is `pi %`.

3 Chunk Options

opts_chunk controls global chunk options, e.g. \texttt{opts_chunk$set(tidy = FALSE)}, which can be overridden by local chunk options. See all options at \texttt{http://yihui.name/knitr/options}; some frequently used options:

\begin{itemize}
\item \texttt{eval} whether to evaluate the chunk
\item \texttt{echo} whether to echo source code
\item \texttt{results} `markup', `asis', `hold', `hide'
\item \texttt{tidy} whether to reformat R code
\item \texttt{cache} whether to cache results
\item \texttt{fig.width}, \texttt{fig.height}, \texttt{out.width}, \texttt{out.height} device and output size of figures
\item \texttt{include} whether to include the chunk results in output
\item \texttt{child} filenames of child documents
\item \texttt{engine} language name (R, python, \ldots)
\end{itemize}

4 Functions

\texttt{knit()} the main function in this package; knit input document and write output

\texttt{purl()} extract R code from an input document

\texttt{spin()} spin goat's hair (an R script with roxygen comments) into wool (a literate programming document to be passed to \texttt{knit()} )

\texttt{stitch()} insert an R script into a template and compile the document

\texttt{knit_hooks$set()} set or reset chunk and output hooks

5 Resources

- homepage: \texttt{http://yihui.name/knitr}
- development repository: \texttt{https://github.com/yihui/knitr} (CRAN, Rforge)
- examples: \texttt{https://github.com/yihui/knitr-examples}
- stackoverflow: \texttt{http://stackoverflow.com/tags/knitr/}
- mailing list: \texttt{https://groups.google.com/group/knitr}