Reproducible Reports with knitr and R Markdown

Yihui Xie, RStudio

11/22/2014 @ UPenn, The Warren Center

An appetizer

Run the app below (your web browser may request access to your microphone).

http://bit.ly/upenn-r-voice

```
install.packages("shiny")
```

Or just use this: https://yihui.shinyapps.io/voice/

Overview and Introduction

I know you click, click, Ctrl+C and Ctrl+V

But imagine you hear these words after you finished a project

Please do that again! (sorry we made a mistake in the data, want to change a parameter, and yada yada)

http://noooooooooooooo.com

Basic ideas of dynamic documents

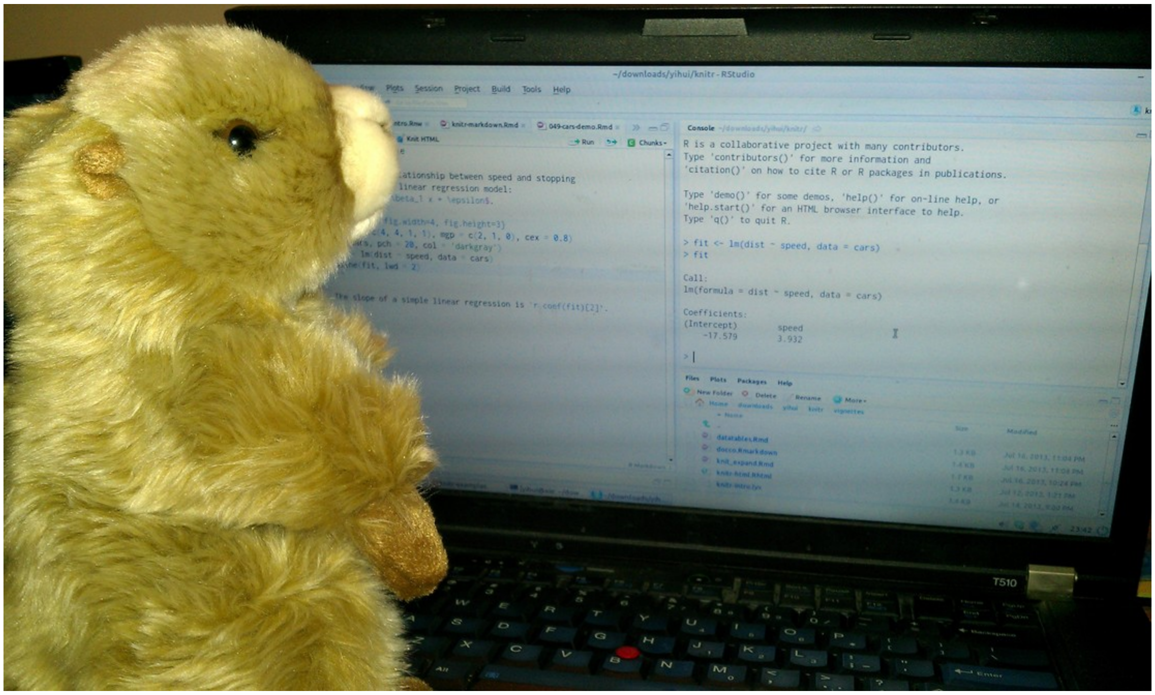
- code + narratives = report
- i.e. computing languages + authoring languages

~~~
We built a linear regression model.

```{r}
fit <- lm(dist ~ speed, data = cars)
b   <- coef(fit)
plot(fit)
```

The slope of the regression is `r b[1]`.
~~~

- an example

Automation! Automation! Automation!

Documentation! Documentation!



~/downloads/yihui/knitr - RStudio
Plots Session Project Build Tools Help
knitr-markdown.Rmd
049-cars-demo.Rmd
Knit HTML
Run
Chunks
Console ~/downloads/yihui/knitr/
Files Plots Packages Help
New Folder Delete Rename More
datatables.Rmd
docco.Rmarkdown
knit_expand.Rmd
T510

```
ationship between speed and stopping
linear regression model:
\beta_1 x + \epsilon$.

fig.width=4, fig.height=3}
c(4, 4, 1, 1), mgp = c(2, 1, 0), cex = 0.8)
ars, pch = 20, col = 'darkgray')
lm(dist ~ speed, data = cars)
ine(fit, lwd = 2)

The slope of a simple linear regression is `r coef(fit)[2]`.
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> fit <- lm(dist ~ speed, data = cars)
> fit

Call:
lm(formula = dist ~ speed, data = cars)

Coefficients:
(Intercept)        speed
    -17.579        3.932

> 
```

Tools

- WEB (Donald Knuth, Literate Programming)
    - Pascal + LaTeX
- Noweb (Norman Ramsey)
- Sweave (Friedrich Leisch and R-core)
    - R + LaTeX
    - extensible, in the sense that you copy 700 lines of code to extend 3 lines
    - cacheSweave, pgfSweave, weaver, …
- knitr (me and contributors)
- odfWeave (Max Kuhn, OpenOffice)
- …

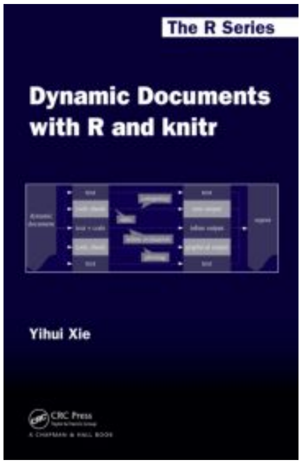
Tools (cont'd)

- Org-mode (Emacs)
- SASweave
- Python tools
  - IPython
  - Dexy
  - PythonTeX

knitr

- an R package (`install.packages('knitr')`)
- document formats
  - .Rnw (R + LaTeX)
  - .Rmd (R + Markdown)
  - any computing language + any authoring language
- editors
  - RStudio, LyX, …
- resources
  - http://yihui.name/knitr
  - Dynamic Documents with R and knitr (Chapman & Hall, 2013)

The knitr book



As an R package

```r
if (!require("knitr")) install.packages("knitr")
library(knitr)
knit("your-document.Rmd")  # compiles a document
```

Minimal examples

- report = text (prose, narrative) + code
- code chunk = chunk header (chunk options) + code
- 02-minimal.Rmd
- 03-minimal.Rnw
- https://github.com/yihui/knitr-examples/
- the RStudio support

knitr Features

Text output

- echo: TRUE/FALSE, c(i1, i2, …), -i3
- results: markup, hide, hold, asis
- collapse: TRUE/FALSE
- warning, error, message
- strip.white
- include
- how to generate tables
- demo: 07-test.Rmd

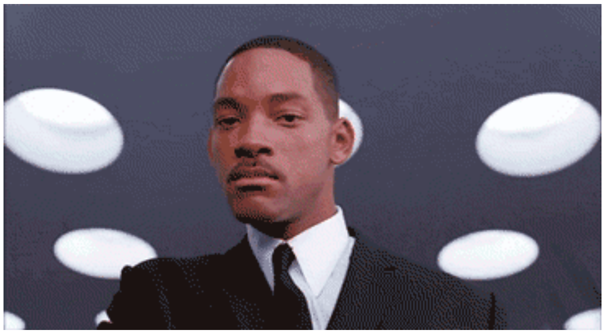
Graphics

- dev, dev.args, fig.ext
    - PDF, PNG, …
    - tikz
- fig.width, fig.height
- out.width, out.height, fig.retina
- fig.cap
- fig.path
- fig.keep
- fig.show
- demo: 08-graphics.Rmd

Caching

- basic idea: use cache if source is not modified
- implementation: digest
- demo: 09-cache.Rmd

Foreign language engines



Foreign language engines

- the chunk option `engine`
- shell scripts
- Python
- Julia (experimental)

```
names(knitr::knit_engines$get())
```

```
##  [1] "awk"       "bash"      "coffee"    "gawk"      "groovy"
##  [6] "haskell"   "node"      "perl"      "python"    "Rscript"
## [11] "ruby"      "sas"       "scala"     "sed"       "sh"
## [16] "zsh"       "highlight" "Rcpp"      "tikz"      "dot"
## [21] "c"         "fortran"   "asy"       "cat"       "asis"
```

- the runr package (experimental)
- demo: 12-python.Rmd, 14-julia.Rmd

R Markdown (v2)

Original Markdown

- primarily for HTML
- paragraphs, `# headers`, `> blockquotes`
- phrase `_emphasis_`
- `- lists`
- `[links](url)`
- `![images](url)`
- code blocks

Pandoc's Markdown

- markdown extensions
  - tables
  - LaTeX math `$\sum_{i=1}^n \alpha_i$` = $\sum_{i=1}^n \alpha_i$
  - YAML metadata
  - raw HTML/LaTeX

    ```
    <div class="my-class">
      ![image](url)
    </div>

    _emphasis_ and \emph{emphasis}
    ```

  - footnotes `^[A footnote here.]`
  - citations `[@joe2014]`

Pandoc's Markdown (cont'd)

- types of output document
    - LaTeX/PDF
    - beamer
    - HTML
    - ioslides
    - reveal.js
    - Word (MS Word, OpenOffice)
    - E-books
    - …

The rmarkdown package

- http://rmarkdown.rstudio.com
- http://www.rstudio.com
- RStudio IDE has included Pandoc binaries
- can be used as a standalone package as well (require separate Pandoc installation)

```r
library(rmarkdown)
render('input.Rmd')
render('input.Rmd', pdf_document())
render('input.Rmd', word_document())
render('input.Rmd', beamer_presentation())
render('input.Rmd', ioslides_presentation())
```

YAML metadata

```
---
title: "Sample Document"
output:
  html_document:
    toc: true
    theme: united
---
```

This is equivalent to:

```
rmarkdown::render('input.Rmd',
    html_document(toc = TRUE, theme = 'united'))
```

What is `html_document()`

```
str(rmarkdown::html_document(), 2)
```

```
## List of 6
##  $ knitr            :List of 3
##   ..$ opts_knit : NULL
##   ..$ opts_chunk:List of 5
##   ..$ knit_hooks: NULL
##  $ pandoc           :List of 5
##   ..$ to      : chr "html"
##   ..$ from    : chr "markdown+autolink_bare_uris+ascii_identifiers
##   ..$ args    : chr [1:8] "--smart" "--email-obfuscation" "none" "
##   ..$ keep_tex: logi FALSE
##   ..$ ext     : NULL
##  $ keep_md          : logi FALSE
##  $ clean_supporting: logi TRUE
##  $ pre_processor   :function (...)
##  $ post_processor  :function (metadata, input_file, output_file, c
##  - attr(*, "class")= chr "rmarkdown_output_format"
```

When in doubt, check out the documentation

- e.g. what are the available options for `html_document`? see

  ```
  ?rmarkdown::html_document
  ```

- http://rmarkdown.rstudio.com

Output format is extensible

```
title: "Sample Document"
output:
  my_nice_document:
    fig_width: 8
    toc: true
```

`my_nice_document()` is your own function that returns a list of **knitr** and Pandoc options. You are recommended to put this function in an R package, and use `MyPKG::my_nice_document` for the `output` field.

Word template?

- see `?rmarkdown::word_document`
- change the styles in a Word document created by Pandoc, and use this document as the "reference document"

RStudio IDE support

You do not have to remember everything

- new markdown document wizard
- setting YAML metadata
- one-click compilation
- navigation between Rmd source and slides

Applications

Websites and blogs

- the UCLA R Tutorial website
    - e.g. http://www.ats.ucla.edu/stat/r/modules/prob_dist.htm
- Jekyll
    - blog like a hacker
    - based on Markdown, supported by Github
    - Rcpp gallery: http://gallery.rcpp.org
- WordPress
    - http://yihui.name/knitr/demo/wordpress/
    - the shinyWP package (under development) https://github.com/yihui/shinyWP

R Package vignettes

- Gentleman and Temple Lang (2004)
- one R package to rule them all!
- `data`, `R`, `man`, `tests`, `demo`, `src`, `vignettes`
- `VignetteBuilder: knitr` in DESCRIPTION
- `\VignetteEngine{knitr::knitr}` in vignettes
- see details

Learn what other cool kids are doing

- courses, blog posts, workshops, papers, R packages, …
- http://yihui.name/knitr/demo/showcase/

Miscellaneous

Reproducible research

- keyword: automation
- the Duke Saga: http://www.economist.com/node/21528593
- not easy for large projects

Repeat: not easy

Markdown or LaTeX?

- LaTeX: precise control, full complexity, horrible readability
- Markdown: simple, simple, simple

```
\section{Introduction}

We did a \emph{cool} study,
and our main findings:


\begin{enumerate}
\item You can never remember
  how to escape backslashes.
\item A dollar sign is \$,
  an ampersand \&, and
  a \textbackslash{}.
\item How about ~? Use $\sin
\end{enumerate}
```

```
# Introduction

We did a _cool_ study, and
our main findings:

1. You do not need to remember
   a lot of rules.
2. A dollar sign is $,
   an ampersand is &, and
   a backslash \.
3. A tilde is ~.

Write content instead of
markup languages.
```

Or a graphical comparison

LaTeX

Markdown

RPubs

- http://rpubs.com
- forget about reproducible research, because you are doing it unconsiously

Is Markdown too simple?

probably, but the real question is

How much do you want?

do you really want *that* word to be in \textbf{\textsf{}}?

To print or not to print, that is the question

- LaTeX is for printing
- HTML is not as powerful in terms of typesetting (not bad either!), but is excellent for interaction
- examples
  - the `docco_classic` style
  - gitbook

The goal

You have done the hard work of research, data collection, and analysis, etc. We hope the last step can be easier.