

Using Rmarkdown and Github for Reproducible Research

Jinliang Yang

November 18, 2014

People behind part of this work



Hadley Wickham

Chief Scientist at RStudio, Inc.

Houston, Texas | Computer Software

Current RStudio, Inc., Rice University

Previous Rice University, University of Auckland

Education Iowa State University



Yihui Xie

Software Engineer at RStudio, Inc.

Des Moines, Iowa Area | Computer Software

Current RStudio, Inc., Supstat Inc

Previous Fred Hutchinson Cancer Research Center, Iowa State University, AT&T Labs, Inc.

Education Iowa State University

A show case of Markdown

- **markdown**: a simple plain text format originally designed for html translation.
The syntax of *markdown* is as simple as only one page.
- **Jekyll**: a static websites generator using **markdown**
see a toy *website* served by *Jekyll*
- Free hosting with **Github pages** with a **gh-pages** branch

A show case of Rmarkdown

- Documents
Create *HTML*, *PDF* and *MS Word* Documents via **RStudio**
- Presentation
Create *Beamer*, *ioslides* and *Slidy* presentations.

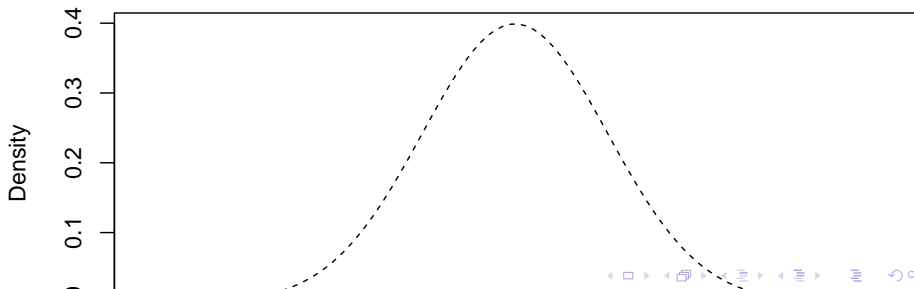
Yihui developed this R package **knitr**, which enables Rmarkdown to include any embedded R code chunks via **Pandoc**.

R code chunks options

- Code evaluation, docoration and plots {r, eval=TRUE, echo=TRUE}

```
x <- seq(-4, 4, length=100)
hx <- dnorm(x)
plot(x, hx, type="l", lty=2, ylab="Density",
     main="A t Distribution")
```

A t Distribution



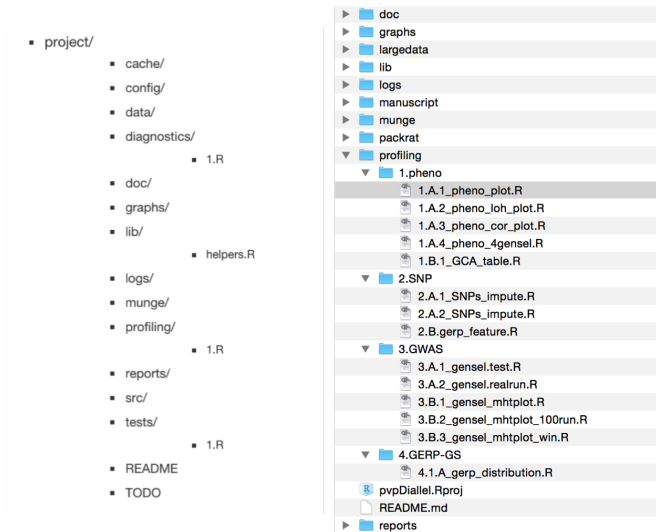
Setup your project

- using **GitHub** to do the version control
- using **packrat** to do the R package dependency management
- using **ProjectTemplate** to layout working directories

```
#install.packages('ProjectTemplate')  
library('ProjectTemplate')  
create.project('temp')  
system("mv temp/* .")  
system("rm -r temp/")
```

Or just copy your favorite directory into the project!

Setup your project



Notes for better reproducible

- Use relative path not absolute path
- **source** your codes rather than embedded as R code chunks
- Turn off the R code evaluation for computational heavy work
- gitignore large data
- for non-R codes, refering **shell** scripts

More notes about general programming

- Small is beautiful.
- Make each program do one thing well.
- Build a prototype as soon as possible.
- Choose portability over efficiency.
- Store data in flat text files.
- Use shell scripts to increase leverage and portability.

By Mike Gancarz (1994)