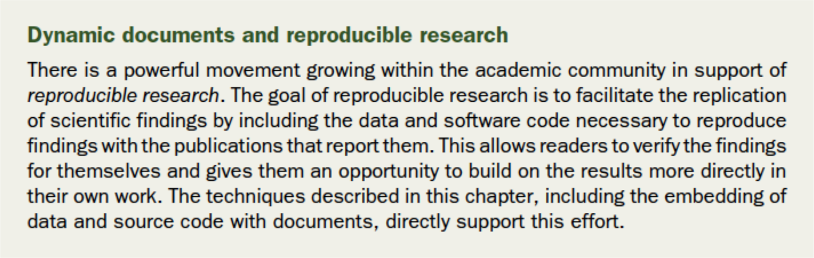
R language and reproducible data analysis

Qiang Shen

Jan. 18, 2017

reproducibility



Dynamic documents and reproducible research

There is a powerful movement growing within the academic community in support of *reproducible research*. The goal of reproducible research is to facilitate the replication of scientific findings by including the data and software code necessary to reproduce findings with the publications that report them. This allows readers to verify the findings for themselves and gives them an opportunity to build on the results more directly in their own work. The techniques described in this chapter, including the embedding of data and source code with documents, directly support this effort.

reproducible research

- **research**:Reproducibile property of research conclusion

reproducible research

- **research**:Reproducibile property of research conclusion
- **researcher**:reproducibility of your own work.

problem

- Run analysis and get the result
- copy paste it into a file and write up the report or paper.

There is no single document to integrate data analysis with textual representations; i.e. data, code, and text are not linked.

problem

- error-prone due to manual work
- tedious jobs to copy and paste
- Graphical User Interface is not recordable
- Tiny change need to redo the whole procedure.
- Communication cost is high for collaboration

Psychology: human vs. computer

- unreliable
- amnesic

How Do I Make My Work Reproducible?

- version control
- literate programming

How Do I Make My Work Reproducible?

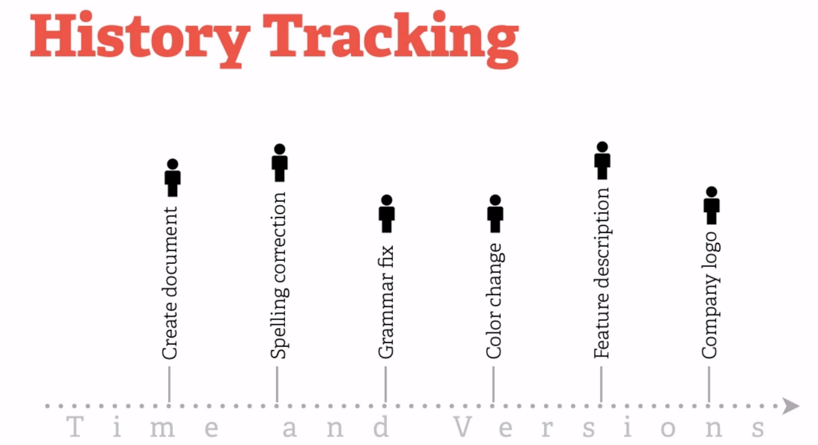
- **version control**
- literate programming

1. Version control

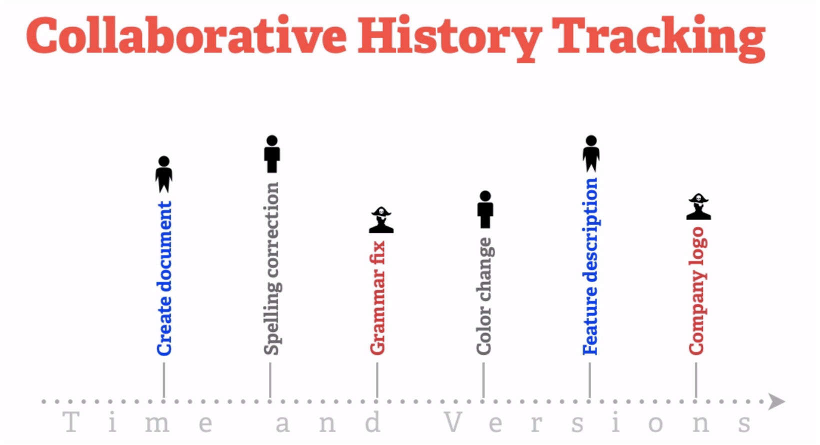
Start from a real scenario: daily tasks

- **Create** things
- **Save** things
- **Edit** things
- Save the thing **again**

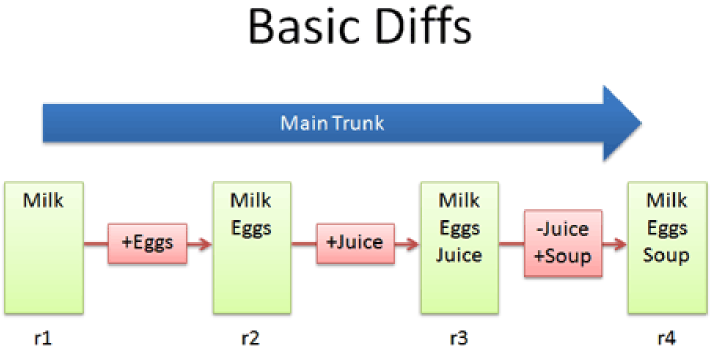
Start from a real scenario



Start from a real scenario



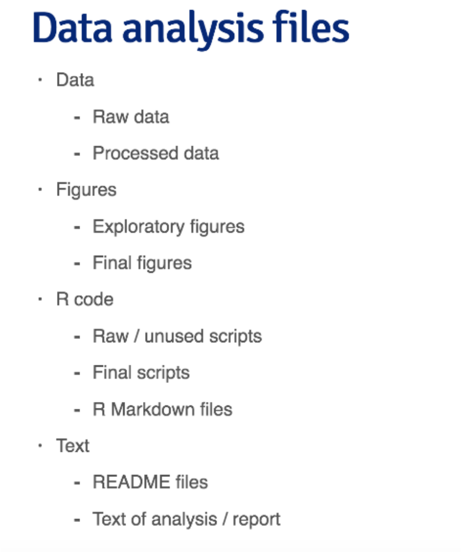
Version control is important!



Cloud storage

- Dropbox
- Nutstore

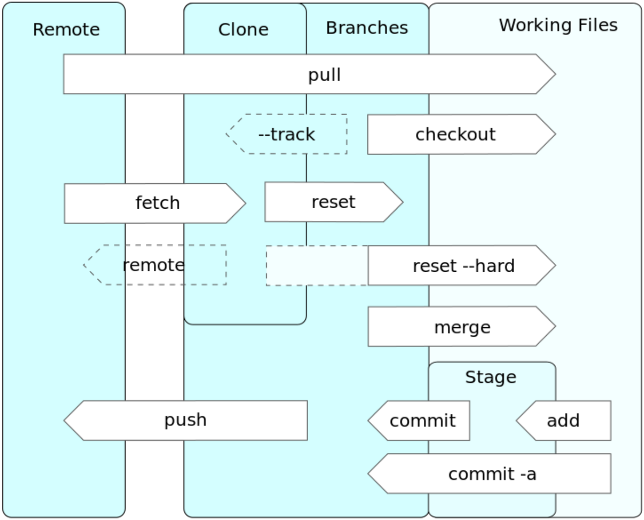
structucture of folder



Data analysis files

- Data
  - Raw data
  - Processed data
- Figures
  - Exploratory figures
  - Final figures
- R code
  - Raw / unused scripts
  - Final scripts
  - R Markdown files
- Text
  - README files
  - Text of analysis / report

Git


Remote
Clone
Branches
Working Files
pull
--track
checkout
fetch
reset
remote
reset --hard
merge
Stage
push
commit
add
commit -a

Github

- a web-based Git repository hosting service

- Bitbucket

Demonstration

- Rstudio
- Github desktop

How Do I Make My Work Reproducible?

- version control
- **literate programming**

2. Literate programming

- conceived by Donald Knuth (Knuth,1984)
- mix the source code and documentation together
- document is divided into text and code “chunks”.
- **weaved** to produce documents and **tangled** to get source code

Literate programming

1. itself is only a concept or idea.
   - A documentation language
   - A programming language
2. **Sweave** system (Friedrich Leisch) used LaTeX and R
3. **knit**r supports a variety of documentation languages

Prerequsites:

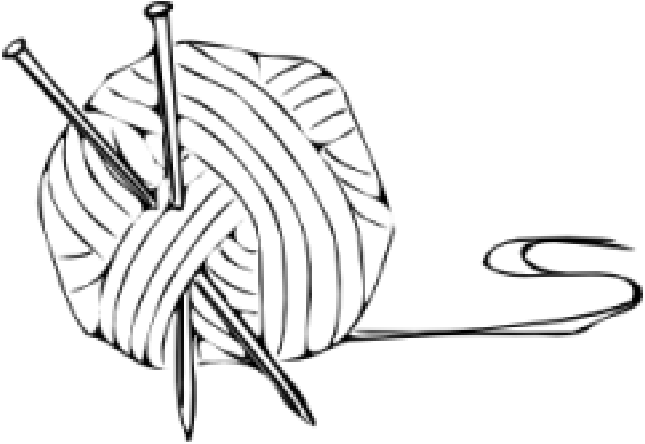
- Basic knowledge of R, Rstudio
- rmarkdown(knitr),xtable, Pandoc
- Latex software,lyx

Latex

1. MiKTEX (Windows: http://miktex.org/),
2. MacTEX (BasicTeX) (Mac OS: http://tug.org/mactex/),
3. TEXLive (Linux: http://tug.org/texlive/).

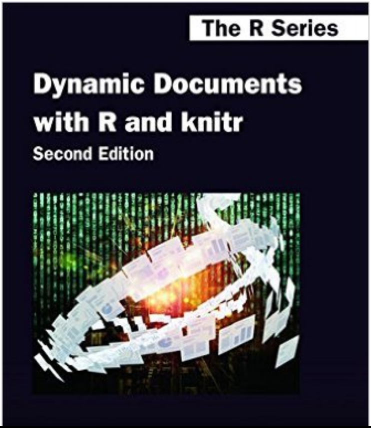
reproducible programming in Rstudio

- Sweave (rstudio->preference->Sweave)
- knitr

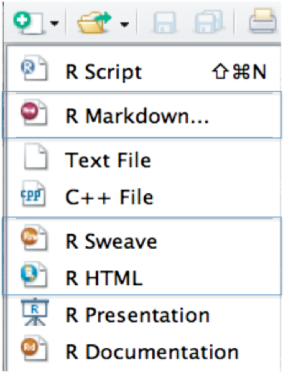


Knitr

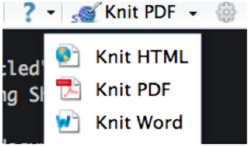
- An R package written by Yihui Xie
- Supports **LaTeX**, **RMarkdown**,and HTML as documentation languages
- Can export to Doc, PDF(slides), HTML
- Built right into RStudio for your convenience.



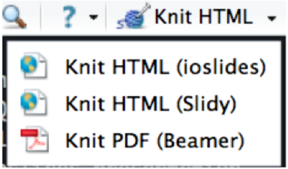
Knitr



Documents



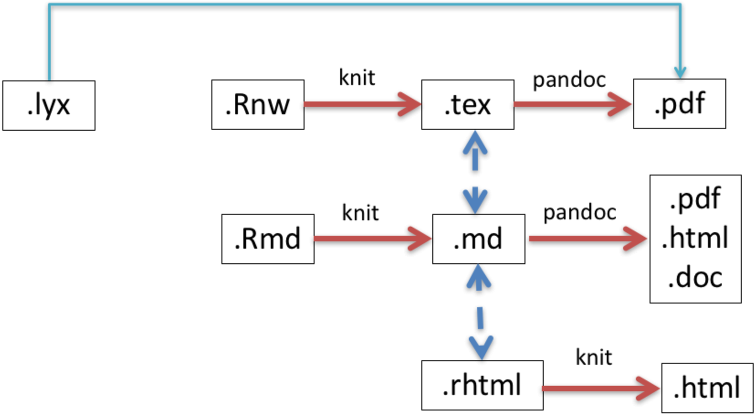
Slides



weave/knit in Rstudio

- Latex
- markdown

framework


.lyx
.Rnw
knit
.tex
pandoc
.pdf
.Rmd
knit
.md
pandoc
.pdf
.html
.doc
.rhtml
knit
.html

Latex

.Rnw in Rstudio

- example-1.Rnw
- example-1-knitr.Rnw
- knitr-minimal.Rnw

http://tobi.oetik-er.ch/lshort/lshort.pdf

lyx

- lyx:https://www.lyx.org/
- compatible with knitr after LyX 2.0.3.

combines the power and flexibility of TeX/LaTeX with the ease of use of a graphical interface.

lyx

- knitr-minimal.lyx
- knitr.lyx

Latex/lyx

R code in .Rnw

- chunks
- inline

```
##chunk
<<>>=
set.seed(1121)
(x=rnorm(20))
mean(x);var(x)
@
##inline
\Sexpr{pi}
```

lyx: table output

```
<<xtable, results="asis">>=
n <- 100
x <- rnorm(n)
y <- 2*x + rnorm(n)
out <- lm(y ~ x)
library(xtable)
xtable(summary(out)$coef, digits=c(0, 2, 2, 1, 2))
@
```

Knitr/lyx

```r
result <- summary(with(mtcars, lm(mpg ~ hp + wt)))
library(knitr)
kable(result$coe)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	37.2272701	1.5987875	23.284689	0.0000000
hp	-0.0317729	0.0090297	-3.518712	0.0014512
wt	-3.8778307	0.6327335	-6.128695	0.0000011

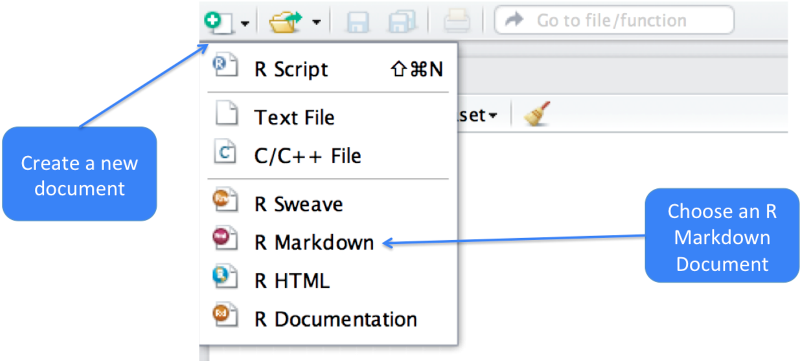
What is markdown

- A simplified version of “markup” languages
- No special editor required
- Simple, intuitive formatting elements

markdown in R: Rmd

- markdown_example.md
- demo.Rmd
- figure.Rmd

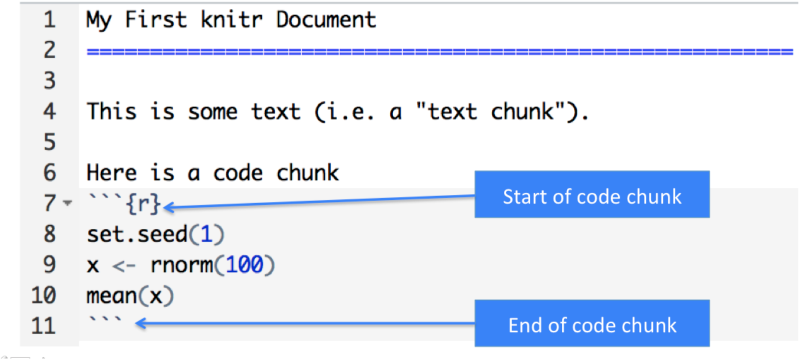
markdown in Rstudio



Create a new document

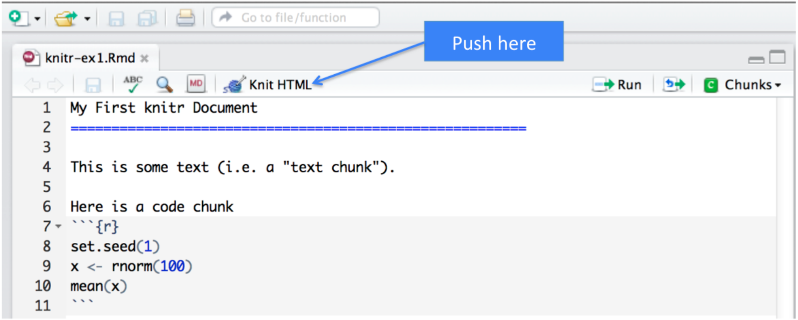
Choose an R Markdown Document

markdown in Rstudio



~~~
My First knitr Document
=======================================================

This is some text (i.e. a "text chunk").

Here is a code chunk
```{r}
set.seed(1)
x <- rnorm(100)
mean(x)
```
~~~

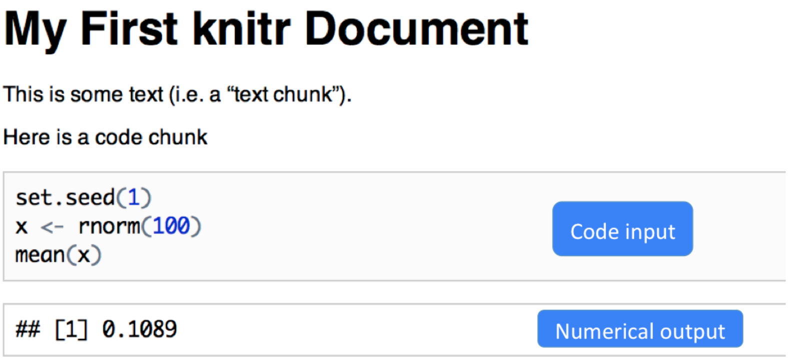
markdown in Rstudio



Push here

~~~
My First knitr Document
========================================================

This is some text (i.e. a "text chunk").

Here is a code chunk
```{r}
set.seed(1)
x <- rnorm(100)
mean(x)
```
~~~

markdown in Rstudio



My First knitr Document

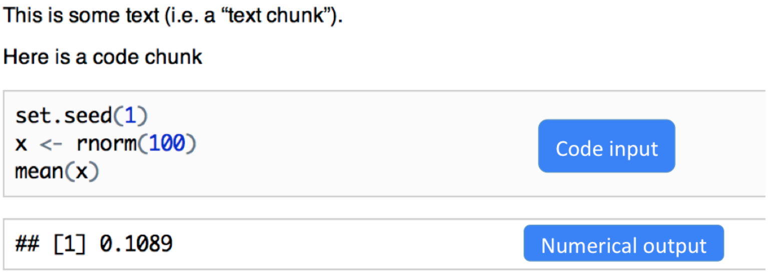
This is some text (i.e. a “text chunk”).

Here is a code chunk

```
set.seed(1)
x <- rnorm(100)
mean(x)
```

```
## [1] 0.1089
```

markdown in Rstudio



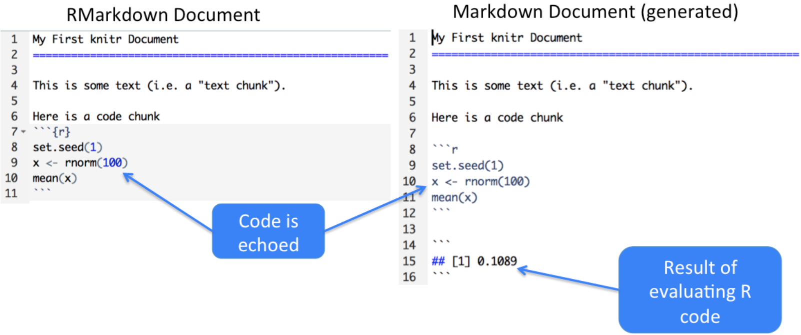
This is some text (i.e. a “text chunk”).

Here is a code chunk

```
set.seed(1)
x <- rnorm(100)
mean(x)
```

```
## [1] 0.1089
```

markdown in Rstudio



RMarkdown Document

~~~
My First knitr Document
=======================================================

This is some text (i.e. a "text chunk").

Here is a code chunk
```{r}
set.seed(1)
x <- rnorm(100)
mean(x)
```
~~~

Markdown Document (generated)

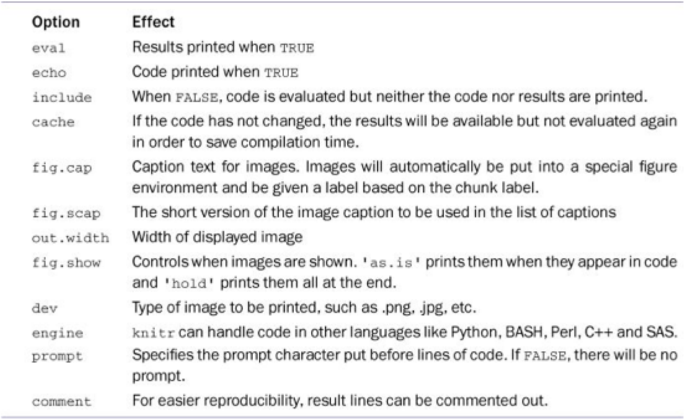
~~~
My First knitr Document
=======================================================

This is some text (i.e. a "text chunk").

Here is a code chunk

```r
set.seed(1)
x <- rnorm(100)
mean(x)
```


```
## [1] 0.1089
```
~~~

Code is echoed

Result of evaluating R code

options

- options
- global options



Option	Effect
`eval`	Results printed when `TRUE`
`echo`	Code printed when `TRUE`
`include`	When `FALSE`, code is evaluated but neither the code nor results are printed.
`cache`	If the code has not changed, the results will be available but not evaluated again in order to save compilation time.
`fig.cap`	Caption text for images. Images will automatically be put into a special figure environment and be given a label based on the chunk label.
`fig.scap`	The short version of the image caption to be used in the list of captions
`out.width`	Width of displayed image
`fig.show`	Controls when images are shown. `'as.is'` prints them when they appear in code and `'hold'` prints them all at the end.
`dev`	Type of image to be printed, such as .png, .jpg, etc.
`engine`	`knitr` can handle code in other languages like Python, BASH, Perl, C++ and SAS.
`prompt`	Specifies the prompt character put before lines of code. If `FALSE`, there will be no prompt.
`comment`	For easier reproducibility, result lines can be commented out.

figures in rmarkdown

```r
n <- 100
x <- rnorm(n)
par(mfrow = c(1, 2), las = 1)
for (i in 1:8) {
    y <- i * x + rnorm(n)
    plot(x, y, main = i)
}
```

figures in rmarkdown

```
![](figure.png)
```

alternative with command

- .Rmd -> .md -> .pdf/.doc/.html
- .Rmd -> .md

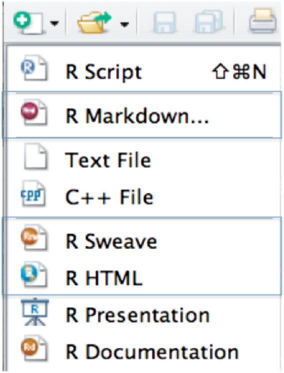
```r
library(knitr)
library(markdown)
## generate .md file
knit("test.Rmd")
```

slidify

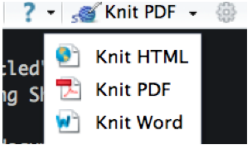
http://slidify.org/start.html

```
# devtools::install_github('slidify', 'ramnathv')
library(slidify)
author("Qiang")
```

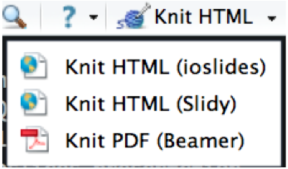
Menu



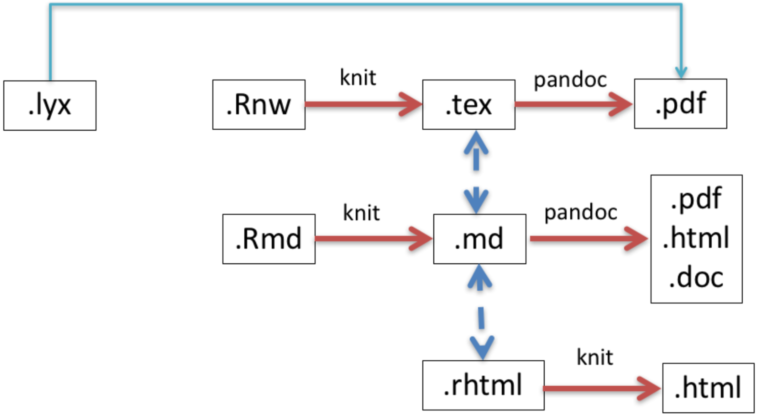
Documents



Slides



Framework


.lyx
.Rnw
knit
.tex
pandoc
.pdf
.Rmd
knit
.md
pandoc
.pdf
.html
.doc
.rhtml
knit
.html

Summary

- version control can be convenient to control your procedure
- Literate programming can be powerful to put text, code, data, output all in one document.
- knitr is a powerful tool for integrating code and text in a simple document format.

Reference

- Roger Peng's coursera course: https://www.coursera.org/instructor/rdpeng
- Yihui Xie's website and book
- R in action 2nd version,chapter 22
- R in data sciences (in Chinese)
- etc.

Q&A

- E-mail:johnsonzhj@gmail.com
- Tel:13675883767