Introduction to markdown and git

Rebecca C. Steorts
Predictive Modeling: STA 521

August 27 2015

Today's Menu

1. What is reproducible research?
2. What is Markdown?
3. What is git and bitbucket?

What is reproducible research?

Reproducible research is the idea that data analyses, and more generally, scientific claims, are published with their data and software code so that others may verify the findings and build upon them.
-Johns Hopkins, Coursera

What is reproducible research?

Reproducible research is still a challenge

Rich FitzJohn Matt Pennell Amy Zanne Will Cornwell

June 9, 2014

Science is reportedly in the middle of a reproducibility crisis. Reproducibility seems laudable and is frequently called for (e.g., nature and science). In general the argument is that research that can be independently reproduced is more reliable than research that cannot be independently reproduced. It is also worth noting that reproducing research is not solely a checking process, and it can provide useful jumping-off points for future research questions. It is difficult to find a counter-argument to these claims, but arguing that reproducibility is laudable in general glosses over the fact that for each research group it is a significant amount of work to make their research (easily) reproducible for independent scientists. While much of the attention has focused on entirely repeating laboratory experiments, there are many simpler forms of reproducibility including, for example, the ability to recompute analyses on known sets of data.

A Case Study

Victoria Stodden

Victoria's Blog

« Data access going the way of journal article access? Insist on open data

Peanut allergic reaction »

Type ar

What the Reinhart & Rogoff Debacle Really Shows: Verifying Empirical Results Needs to be Routine

Published on April 19, 2013 in Uncategorized. 3 Comments

There's been an enormous amount of buzz since a study was released this week questioning the methodology in a published paper. The paper under fire is Reinhart and Rogoff's "Growth in a Time of Debt" and the firing is being done by Herndon, Ash, and Pollin in their article "Does High Public Debt Consistently Stifle Economic Growth? A Critique of Reinhart and Rogoff." Herndon, Ash, and Pollin claim to have found "spreadsheet errors, omission of available data, weighting, and transcription" in the original research which, when corrected, significantly reduce the magnitude of the original findings. These corrections were possible because of openness in economics, and this openness needs to be extended to make all computational publications reproducible.

A Case Study

In 1986 a study was published in The American Economic Review (the very same journal that published Reinhart and Rogoff's piece) by Dewald, Thursby, and Anderson called "Replication in Empirical Economics: The Journal of Money, Credit and Banking Project" detailing shocking results: of 152 papers published or to be published in the Journal of Money, Credit and Banking between 1982–1984, only 4 could essentially be replicated in their entirety (the authors only received sufficient data and code for 9). The reason the JMCB was selected for study was their novel (at the time) policy of requiring all authors to relinquish data and code, to be made available to other researchers. These stark results sent shockwaves through the economics community and many journals, including the AER, subsequently implemented data and code release requirements. Economists also became more aware of the issue of reproducibility and many do release the data and code associated with their published studies.

- How did such papers pass peer review?
- How would you reproduce such results without their data or their code?
- Suggestion: At the time of publication, researchers make enough material openly available (data, programs, narrative) so that other researchers in the field can replicate their work.
- At the very least, this sets a standard not just for academia but for things in house in industry.

Research and Education

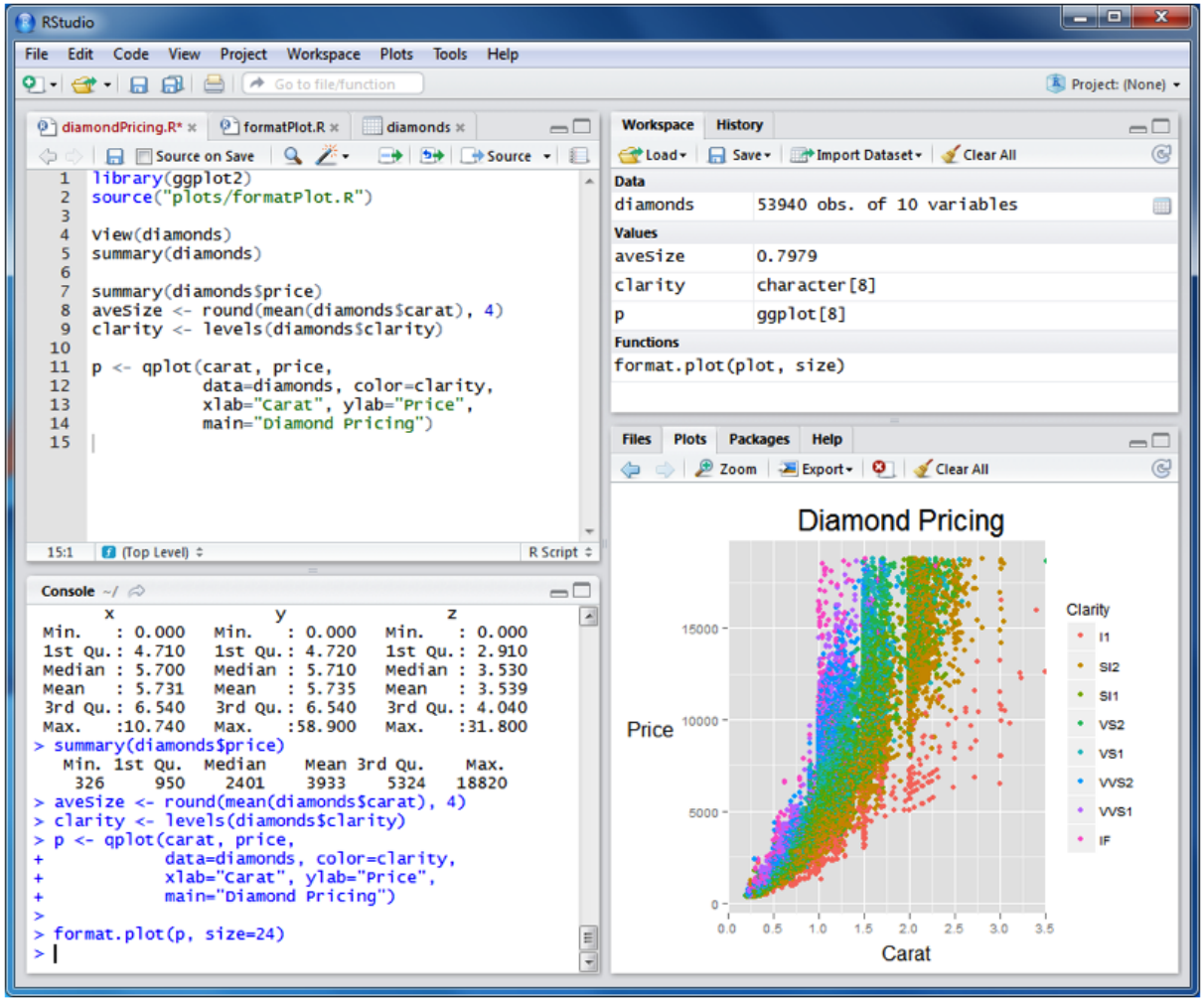
how to organize your work?
how to make work more pleasant for you?
how to make it navigable by others?
how to reduce tedium and manual processes?
how to reduce friction for collaboration?
how to reduce friction for communication?

specific tools and habits can build alot of this into the normal coding and analysis process

[Credit: Jenny Bryan]

RStudio

RStudio is an integrated development environment (IDE) for R



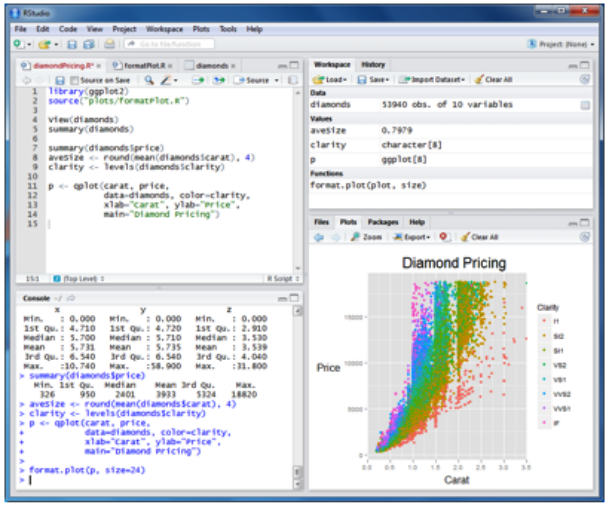
[Credit: Jenny Bryan]

RStudio

R ≠ RStudio

RStudio mediates your interaction with R; it would replace Emacs + ESS or Tinn-R, but not R itself

Rstudio is a product of -- actually, more a driver of -- the emergence of R Markdown, `knitr`, R + Git(Hub)



[Credit: Jenny Bryan]

markdown

Markdown

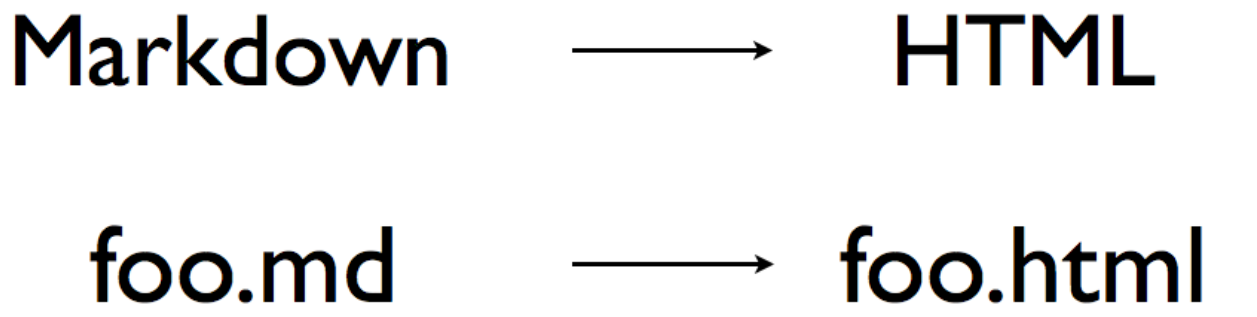
What is Markdown?

- Markdown is a lightweight markup language for creating HTML (or XHTML) documents.
- Markup languages are designed produce documents from human readable text (and annotations).
- Some of you may be familiar with *LaTeX*. This is another (less human friendly) markup language for creating pdf documents.
- Why I love Markdown:
  - Easy to learn and use.
  - Focus on **content**, rather than **coding** and debugging **errors**.
  - Once you have the basics down, you can get fancy and add HTML, JavaScript & CSS.

http://cpsievert.github.io/slides/markdown/#/5

[Credit: Jenny Bryan]

Markdown

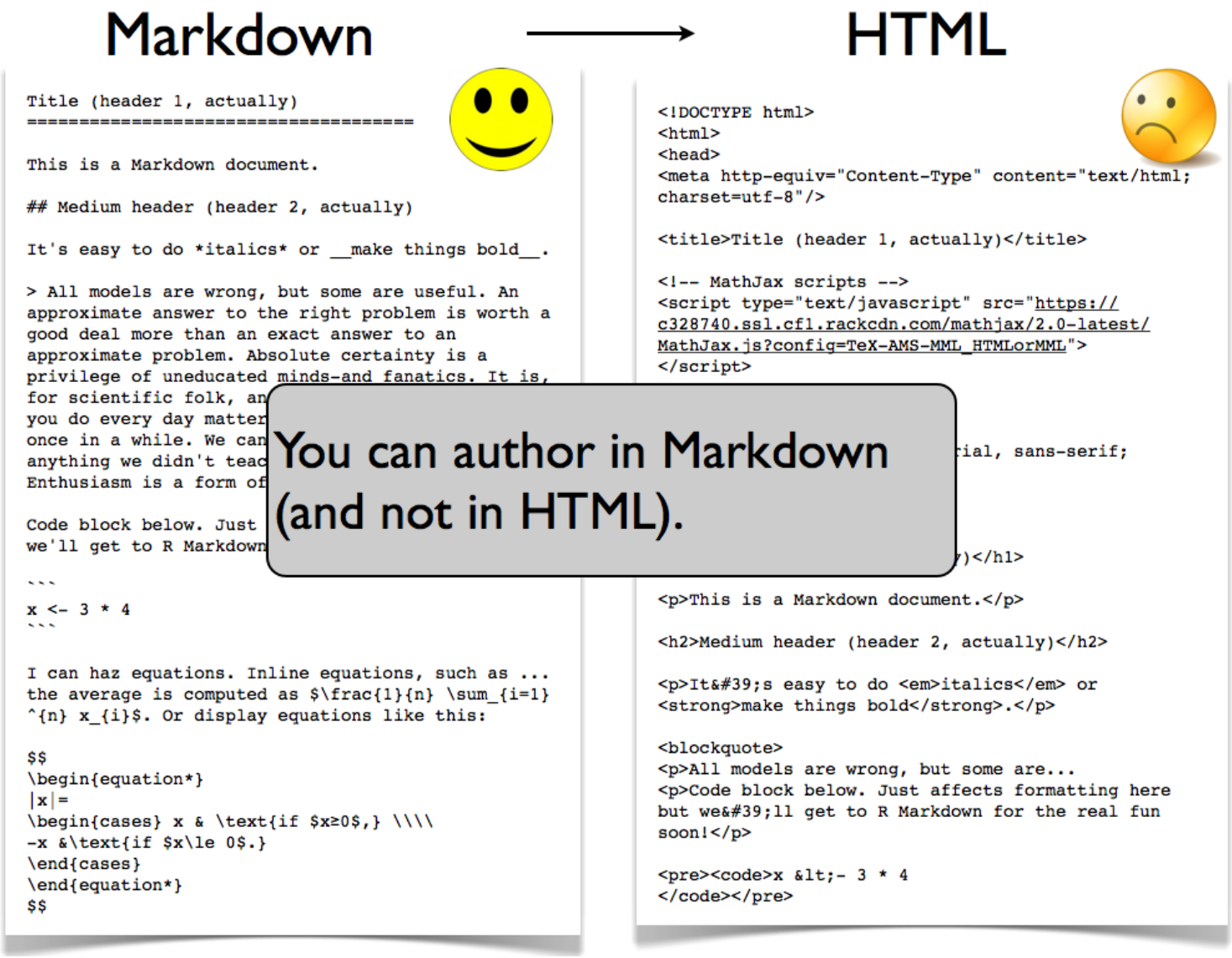


easy to write
(and read!)

easy to publish
easy to read in
browser

[Credit: Jenny Bryan]

Markdown



[Credit: Jenny Bryan]

Markdown

Markdown → HTML

```
Title (header 1, actually)
==================================

This is a Markdown document.

## Medium header (header 2, actually)

It's easy to do *italics* or __make things bold__.

> All models are wrong, but some are useful. An
approximate answer to the right problem is worth a
good deal more than an exact answer to an
approximate problem. Absolute certainty is a
privilege of uneducated minds-and fanatics. It is,
for scientific folk, an unattainable ideal. What
you do every day matters more than what you do
once in a while. We cannot expect anyone to know
anything we didn't teach them ourselves.
Enthusiasm is a form of social courage.

Code block below. Just affects formatting here but
we'll get to R Markdown for the real fun soon!

```
x <- 3 * 4
```

I can haz equations. Inline equations, such as ...
the average is computed as $\frac{1}{n} \sum_{i=1}
^{n} x_{i}$. Or display equations like this:

$$
\begin{equation*}
|x|=
\begin{cases} x & \text{if $x\ge0$,} \\\\
-x &\text{if $x\le 0$.}
\end{cases}
\end{equation*}
$$
```

Title (header 1, actually)

This is a Markdown document.

Medium header (header 2, actually)

It's easy to do *italics* or **make things bold**.

> All models are wrong, but some are useful. An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem. Absolute certainty is a privilege of uneducated minds-and fanatics. It is, for scientific folk, an unattainable ideal. What you do every day matters more than what you do once in a while. We cannot expect anyone to know anything we didn't teach them ourselves. Enthusiasm is a form of social courage.

Code block below. Just affects formatting here but we'll get to R Markdown for the real fun soon!

```
x <- 3 * 4
```

I can haz equations. Inline equations, such as ... the average is computed as $\frac{1}{n} \sum_{i=1}^{n} x_i$. Or display equations like this:

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x \leq 0. \end{cases}$$

[Credit: Jenny Bryan]

Pandoc

If I use Markdown, am I restricted to HTML output?

No.

pandoc = "swiss-army knife" of document conversion

(RStudio will gladly install and invoke for you.)

About pandoc

If you need to convert files from one markup format into another, pandoc is your swiss-army knife. Pandoc can convert documents in markdown, reStructuredText, textile, HTML, DocBook, LaTeX, MediaWiki markup, OPML, or Haddock markup to

- HTML formats: XHTML, HTML5, and HTML slide shows using Slidy, reveal.js, Slideous, S5, or DZSlides.
- Word processor formats: Microsoft Word docx, OpenOffice/LibreOffice ODT, OpenDocument XML
- Ebooks: EPUB version 2 or 3, FictionBook2
- Documentation formats: DocBook, GNU TexInfo, Groff man pages, Haddock markup
- Outline formats: OPML
- TeX formats: LaTeX, ConTeXt, LaTeX Beamer slides
- PDF via LaTeX
- Lightweight markup formats: Markdown, reStructuredText, AsciiDoc, MediaWiki markup, Emacs Org-Mode, Textile
- Custom formats: custom writers can be written in lua.

[Credit: Jenny Bryan]

Markdown

If you have an annoying process for authoring for the web ....

or

If you avoid authoring for the web, because you're not sure how ...

start writing in Markdown.

[Credit: Jenny Bryan]

R markdown

You can include bits of `R` code in Markdown. Let's see how it works!

R Markdown

R Markdown

````
R Markdown rocks
====================================

This is an R Markdown document.

```{r}
x <- rnorm(1000)
head(x)
```

See how the R code gets executed and a
representation thereof appears in the document?
`knitr` gives you control over how to represent all
conceivable types of output. In case you care, then
average of the `r length(x)` random normal variates
we just generated is `r round(mean(x), 3)`. Those
numbers are NOT hard-wired but are computed on-the-
fly. As is this figure. No more copy-paste ... copy-
paste ... oops forgot to copy-paste.

```{r}
plot(density(x))
```

Note that all the previously demonstrated math
typesetting still works. You don't have to choose
between having math cred and being web-friendly!

Inline equations, such as ... the average is
computed as $\frac{1}{n} \sum_{i=1}^{n} x_{i}$. Or
display equations like this:

$$
\begin{equation*}
|x|=
\begin{cases} x & \text{if $x\ge0$,} \\\\
-x &\text{if $x\le 0$.}
\end{cases}
\end{equation*}
$$
````

→

Markdown

````
R Markdown rocks
====================================

This is an R Markdown document.

```r
x <- rnorm(1000)
head(x)
```

```
## [1] -1.3007  0.7715  0.5585 -1.2854  1.1973
2.4157
```

See how the R code gets executed and a
representation thereof appears in the document?
`knitr` gives you control over how to represent all
conceivable types of output. In case you care, then
average of the 1000 random normal variates we just
generated is -0.081. Those numbers are NOT hard-
wired but are computed on-the-fly. As is this
figure. No more copy-paste ... copy-paste ... oops
forgot to copy-paste.

```r
plot(density(x))
```

![plot of chunk unnamed-chunk-2](figure/unnamed-
chunk-2.png)

...
````

R Markdown

Markdown ⟶ HTML

````
R Markdown rocks
==================================

This is an R Markdown document.


```r
x <- rnorm(1000)
head(x)
```


```
## [1] -1.3007  0.7715  0.5585 -1.2854  1.1973
2.4157
```


See how the R code gets executed and a
representation thereof appears in the document?
`knitr` gives you control over how to represent all
conceivable types of output. In case you care, then
average of the 1000 random normal variates we just
generated is -0.081. Those numbers are NOT hard-
wired but are computed on-the-fly. As is this
figure. No more copy-paste ... copy-paste ... oops
forgot to copy-paste.


```r
plot(density(x))
```


![plot of chunk unnamed-chunk-2](figure/unnamed-
chunk-2.png)

...
````

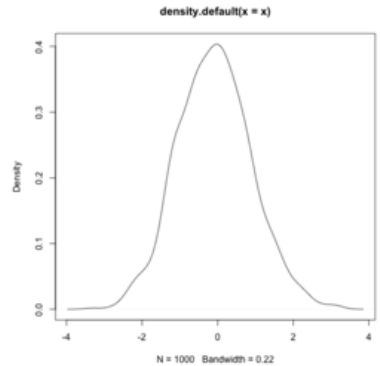
R Markdown rocks

This is an R Markdown document.

```
x <- rnorm(1000)
head(x)
```

```
## [1] -1.3007  0.7715  0.5585 -1.2854  1.1973  2.4157
```

See how the R code gets executed and a representation thereof appears in the document? `knitr` gives you control over how to represent all conceivable types of output. In case you care, then average of the 1000 random normal variates we just generated is -0.081. Those numbers are NOT hard-wired but are computed on-the-fly. As is this figure. No more copy-paste … copy-paste … oops forgot to copy-paste.

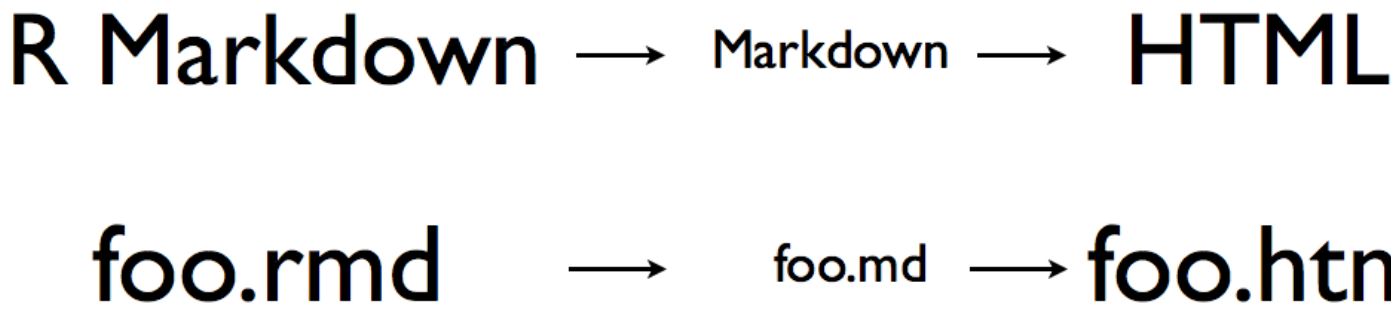


Note that all the previously demonstrated math typesetting still works. You don't have to choose between having math cred and being web-friendly!

Inline equations, such as … the average is computed as $\frac{1}{n}\sum_{i=1}^{n} x_i$. Or display equations like this:

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x \leq 0. \end{cases}$$

R Markdown



easy to write
(and read!)

easy to publish
easy to read in
browser

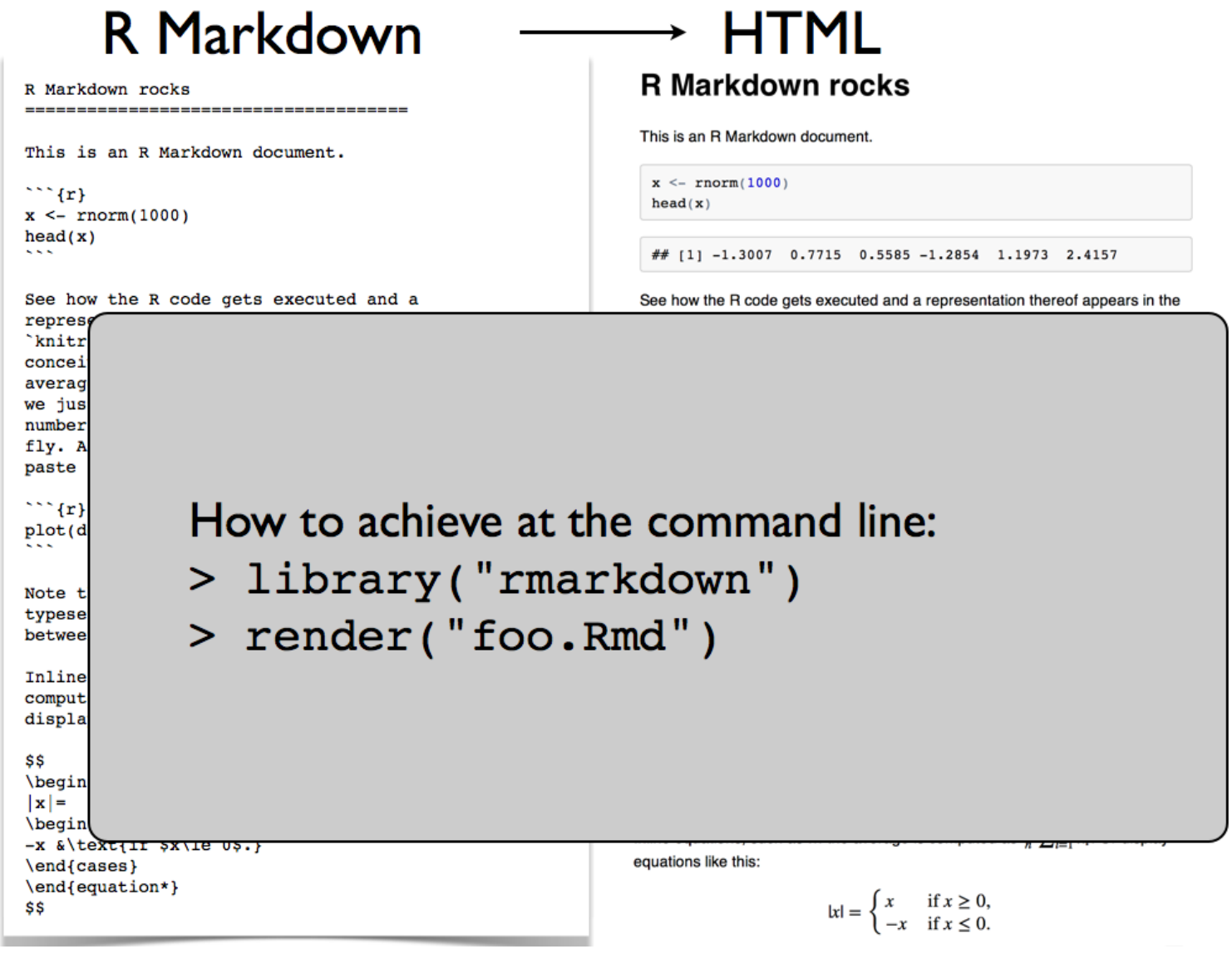
R Markdown

How do to actually convert Markdown to HTML?

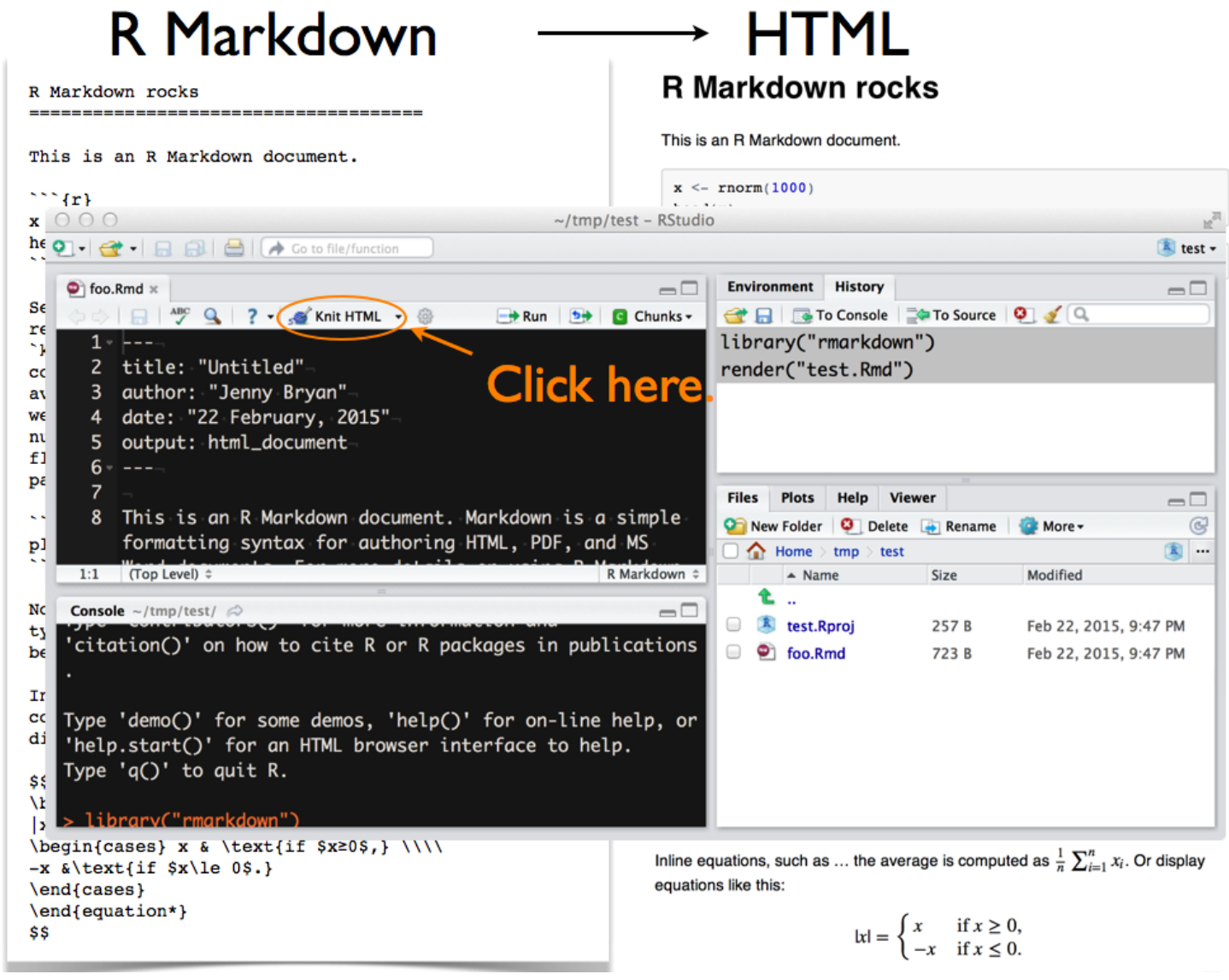
`knitr`, `rmarkdown` add-on packages provide user-friendly functions

RStudio makes them available via button

R Markdown


R Markdown
HTML
R Markdown rocks
This is an R Markdown document.
x <- rnorm(1000)
head(x)
[1] -1.3007 0.7715 0.5585 -1.2854 1.1973 2.4157
See how the R code gets executed and a representation thereof appears in the
equations like this:
How to achieve at the command line:
> library("rmarkdown")
> render("foo.Rmd")

R Markdown


R Markdown
HTML
R Markdown rocks
This is an R Markdown document.
~/tmp/test – RStudio
Knit HTML
Run
Chunks
Click here.
Environment
History
To Console
To Source
library("rmarkdown")
render("test.Rmd")
Files
Plots
Help
Viewer
New Folder
Delete
Rename
More
Home
tmp
test
Name
Size
Modified
test.Rproj
257 B
Feb 22, 2015, 9:47 PM
foo.Rmd
723 B
Feb 22, 2015, 9:47 PM
Console ~/tmp/test/
'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.
> library("rmarkdown")
Inline equations, such as ... the average is computed as $\frac{1}{n}\sum_{i=1}^{n} x_i$. Or display equations like this:
$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x \leq 0. \end{cases}$$

R Markdown

Do I have to do everything in R markdown? What about plain R scripts?

Use `rmarkdown::render()` or Rstudio's Compile Notebook button to get a satisfying stand-alone webpage based on an R script.

R Markdown

simple R script:
`toyline.R`

```r
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))

(avgX <- mean(x))

plot(x, y)
abline(a, b, col = "blue", lwd = 2)
```

→ HTML

toyline.R

jenny — *Sep 6, 2013, 3:15 PM*

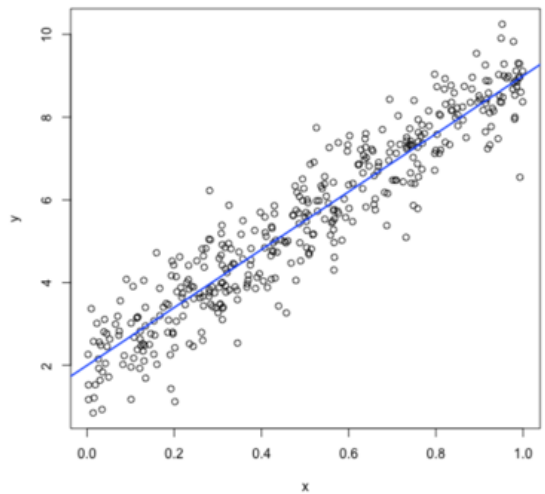
```r
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

set.seed(1234)
x <- runif(n)
y <- a + b * x + rnorm(n, sd = sqrt(sigSq))

(avgX <- mean(x))
```

```
[1] 0.4969
```

```r
plot(x, y)
abline(a, b, col = "blue", lwd = 2)
```



R Markdown

How do I show the world all these awesome dynamic HTML reports I'm creating?

Easiest: Rpubs

Or do whatever you usually do to get HTML on the web.

Or use GitHub ....

R Markdown

Summary:

web-friendly is good

various hosting platforms make it easy to share web-ready products with minimal effort

embedding analysis and logic in source document for a report is good
- huge win for reproducibility
- also excellent for communication and documentation

(R) Markdown + `knitr` (+ RStudio) make it very easy to author dynamic reports that are ready for the web

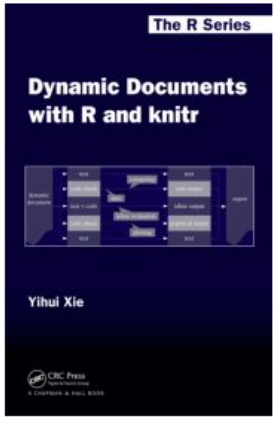
R Markdown

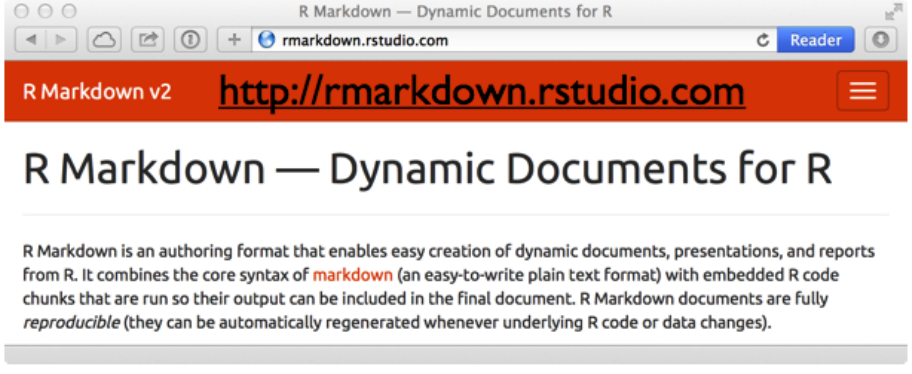
disclaimer:

knitr is **not limited** to executing R code
knitr is **not limited** to processing R Markdown

I just chose to focus on R and R Markdown

Read more in the book or on the web:
Dynamic documents with R and knitr by Yihui Xie, part of the CRC Press / Chapman & Hall R Series (2013). ISBN: 9781482203530.





Homework 1

- Your first homework can be found at `https://stat.duke.edu/~rcs46/labsANDhw.html`
- Note that you must complete this in Markdown and you must submit this through the Sakai website.
- The homework is due Monday, August 31 at 11:59 PM.
- Over the weekend, please also install bitbucket and git (using the instructions below).
- Please also complete the reading on git. There will be no reading for Tuesday's class.

Sometimes we need a little organization!

Version Control

Version control are tools that allow individuals or groups to work simultaneously on the same project.

- Mastering a version control system is vital to easily collaborate with others, and is useful even for solo work because it allows you to easily undo mistakes.
- You can use it in combo with RStudio. For more, see `http://r-pkgs.had.co.nz/intro.html`.

Git

Git is a version control system.

- It's a tool that tracks changes to your code and shares those changes with others.
- Git is most useful when combined with GitHub or Bitbucket, a website that allows you to share your code with the world, solicit improvements via pull requests and track issues.

Why Git + Git (Bitbucket)

- Sharing packages and programs is easy. Any R user can install your package via:

```
install.packages("devtools")
devtools::install_github("username/packagename")
```

- It's a great way to maintain reproducible research that others can report suggestions or bugs on.
- It's nice for collaborative or team projects so you don't have to share code via Dropbox or email.
- You can also undo and spot mistakes easily and track these down.

Basic set up and commands

We'll now move to my webpage so we can look at basic set up and commands. `https://stat.duke.edu/~rcs46/git.html`