literate programming

prepared by Jenny Bryan for

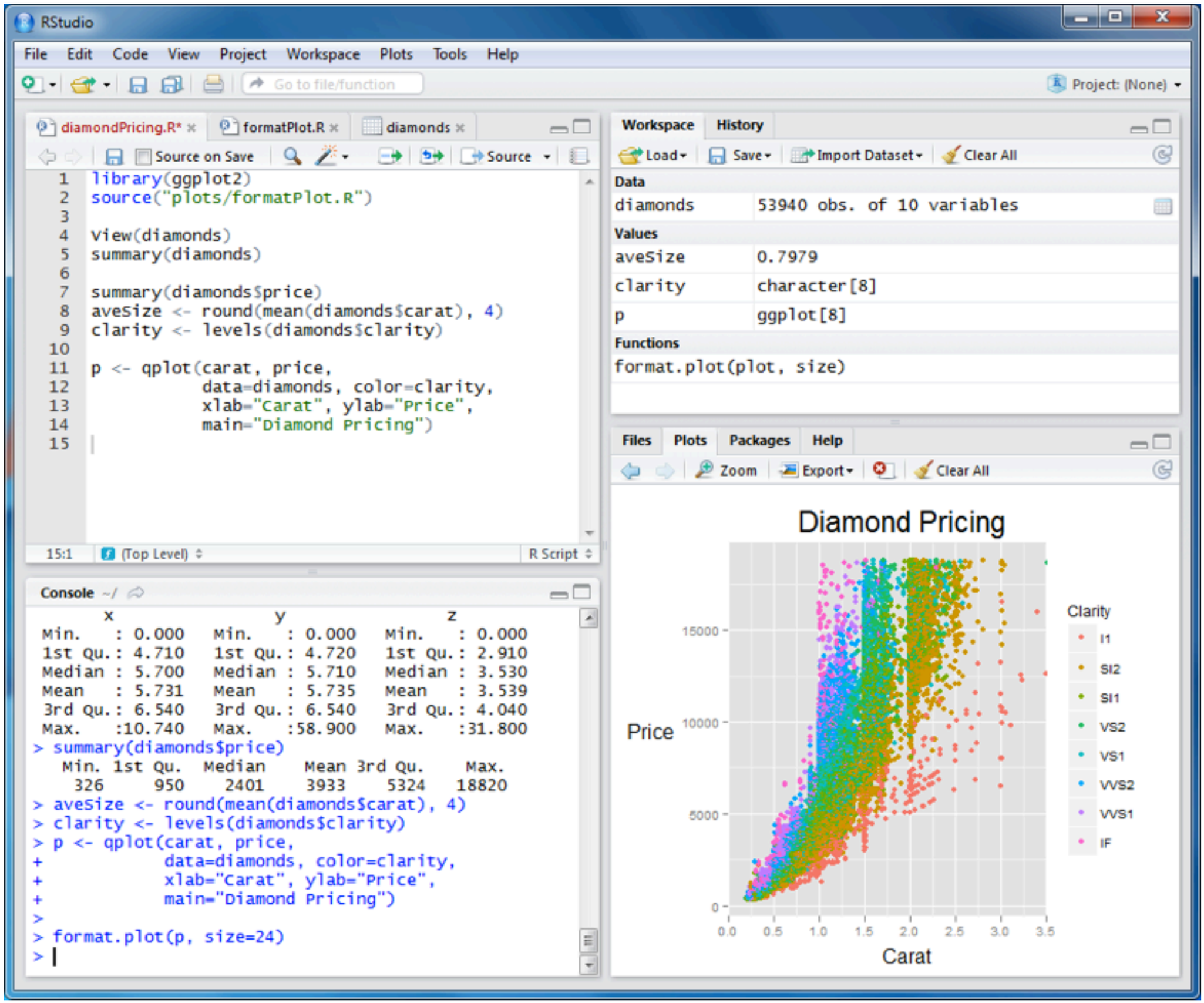
Reproducible Science Workshop

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

weak links in the chain: process, packaging and presentation

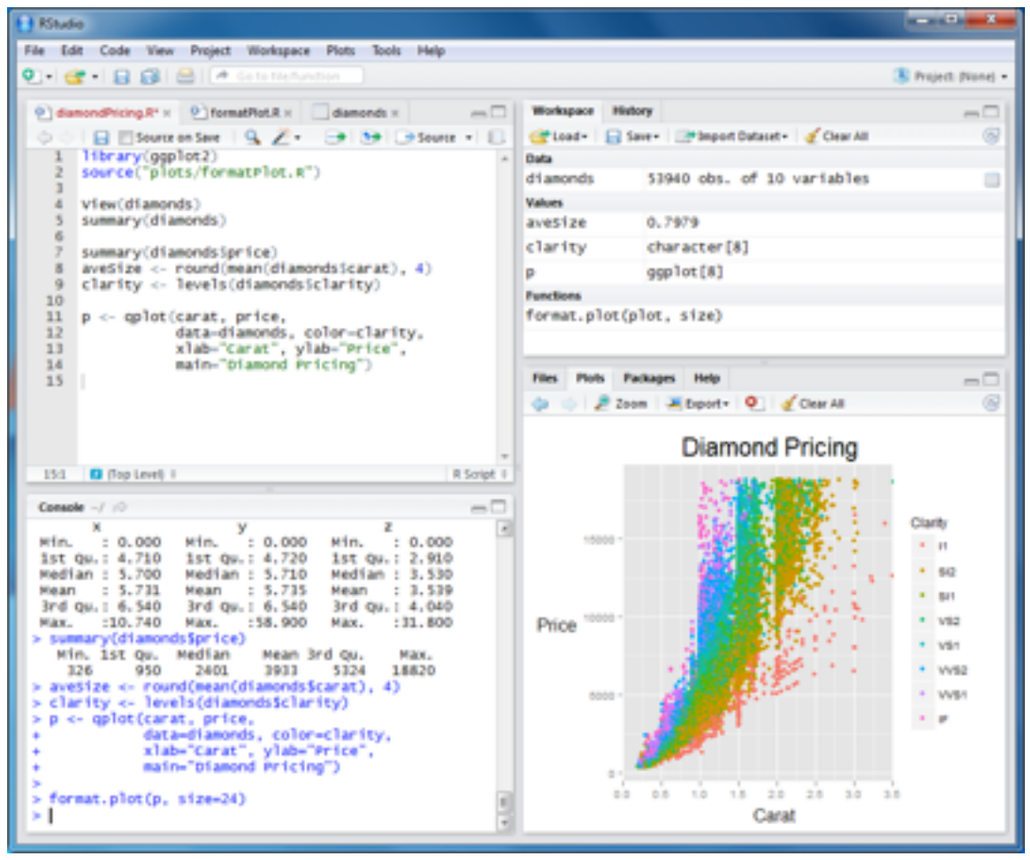
RStudio is an integrated development environment (IDE) for R



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)



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

Markdown → HTML

foo.md → foo.html

easy to write
(and read!)

easy to publish
easy to read in
browser

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
you do every day matter
once in a while. We can
anything we didn't teac
Enthusiasm is a form of

Code block below. Just
we'll get to R Markdown

```
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:

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

You can author in Markdown (and not in HTML).

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8"/>

<title>Title (header 1, actually)</title>

<!-- MathJax scripts -->
<script type="text/javascript" src="https://
c328740.ssl.cf1.rackcdn.com/mathjax/2.0-latest/
MathJax.js?config=TeX-AMS-MML_HTMLorMML">
</script>

                                    rial, sans-serif;

                                    y)</h1>

<p>This is a Markdown document.</p>

<h2>Medium header (header 2, actually)</h2>

<p>It&#39;s easy to do <em>italics</em> or
<strong>make things bold</strong>.</p>

<blockquote>
<p>All models are wrong, but some are...
<p>Code block below. Just affects formatting here
but we&#39;ll get to R Markdown for the real fun
soon!</p>

<pre><code>x &lt;- 3 * 4
</code></pre>
```

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:

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

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:

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

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.

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.

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:

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

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)

...
````

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)

...
````

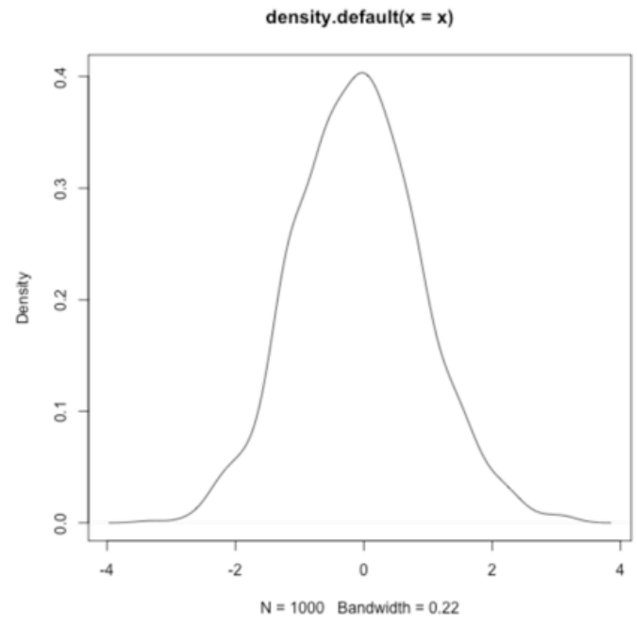
R Markdown rocks

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 → Markdown → HTML

foo.rmd → foo.md → foo.html

easy to write
(and read!)

easy to publish
easy to read in
browser

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 → HTML

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

This is an R Markdown document.

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

See how the R code gets executed and a
repres
`knitr
concei
averag
we jus
number
fly. A
paste

```{r}
plot(d
```

Note t
typese
betwee

Inline
comput
displa

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

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

How to achieve at the command line:

```
> library("rmarkdown")
> render("foo.Rmd")
```

equations like this:

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

R Markdown ⟶ HTML

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

This is an R Markdown document.

```{r}
x
he
```

Se
re
`k
co
av
we
nu
fl
pa

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

R Markdown rocks

This is an R Markdown document.

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

~/tmp/test – RStudio

Go to file/function

test

foo.Rmd

Knit HTML | Run | Chunks

```
1  ---
2  title: "Untitled"
3  author: "Jenny Bryan"
4  date: "22 February, 2015"
5  output: html_document
6  ---
7
8  This is an R Markdown document. Markdown is a simple
   formatting syntax for authoring HTML, PDF, and MS
```

1:1 (Top Level) R Markdown

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:

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

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.

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*

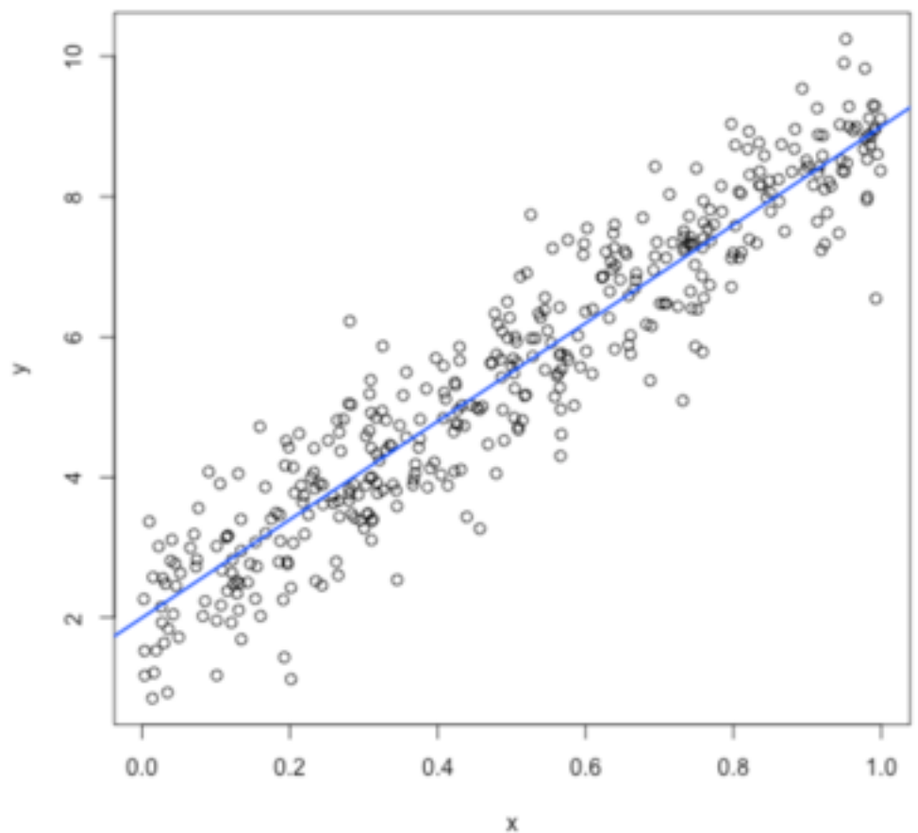
```r
a <- 2
b <- 7
sigSq <- 0.5
n <- 400

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)
```



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 ....

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

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.