

Literate Programming in R Markdown

David A. Selby

Department of Statistics, University of Warwick

16 November 2016

- ① Literate Programming
- ② Markdown
- ③ R Markdown
- ④ Lazy, productive research

Literate Programming

Motivation

- 1 Literate programming helps peers understand and replicate your results, find errors and suggest enhancements

Motivation

- 1 Literate programming helps peers understand and replicate your results, find errors and suggest enhancements
- 2 “Literate programming produces better-quality programs” — *Donald Knuth*

Motivation

- 1 Literate programming helps peers understand and replicate your results, find errors and suggest enhancements
- 2 “Literate programming produces better-quality programs” — *Donald Knuth*
- 3 Literate programming saves time and effort, so you can spend more time:

Motivation

- 1 Literate programming helps peers understand and replicate your results, find errors and suggest enhancements
- 2 “Literate programming produces better-quality programs” — *Donald Knuth*
- 3 Literate programming saves time and effort, so you can spend more time:
 - doing *real* research

Motivation

- 1 Literate programming helps peers understand and replicate your results, find errors and suggest enhancements
- 2 “Literate programming produces better-quality programs” — *Donald Knuth*
- 3 Literate programming saves time and effort, so you can spend more time:
 - doing *real* research
 - in the pub

Effective communication

“If you can’t write clearly, you probably don’t think nearly as well as you think you do.” — Kurt Vonnegut

*“If it was hard to write, it should be hard to read.”
— Computer programmers’ proverb*

Commenting code

What does this code do?

```
data(women)
plot(women)
fit <- lm(weight ~ height, data = women)
abline(fit)
```

Commenting code

With comments:

```
# Analysis of the 'women' dataset in R  
data(women) # Load the data  
plot(weight~height, data = women) # Make a scatter plot  
fit <- lm(weight ~ height, data = women) # Fit linear model  
abline(fit) # Add a line of best fit to the plot
```

Literate Programming

“Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to humans what we want the computer to do.”

— Donald Knuth

Who will read your code?

- 1 Your supervisor
- 2 Collaborators
- 3 Reviewers
- 4 Future *you*

The *World Almanac and Book of Facts* (1975) includes a dataset of heights (in) and weights (lbs) of 15 American women aged 30–39. It is built into R:

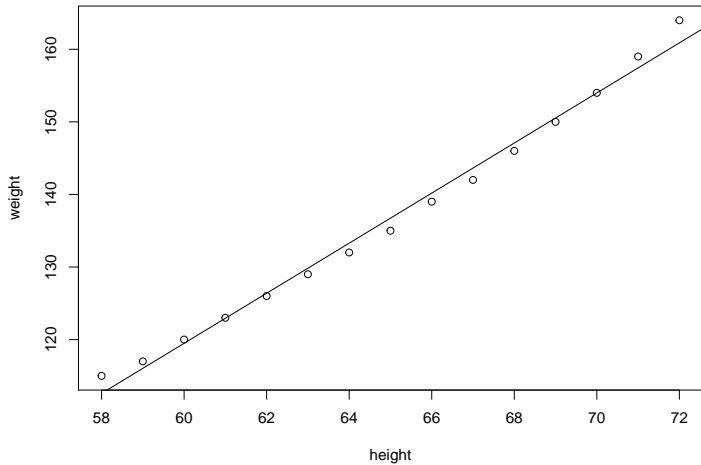
```
data(women)
```

As height increases, weight appears to increase (almost) linearly: every inch in height adds approximately 3.45 lbs. This was determined by fitting a simple linear regression model of weight against height:

```
fit <- lm(weight ~ height, data = women)
```

The resulting least-squares regression line can be drawn on a scatter plot of height against weight. The fit looks quite good...

```
plot(weight~height, data = women)  
abline(fit)
```



Markdown

Markdown syntax

Here is some text in **italics**, in ****bold**** and ``teletype``.

Here is a new paragraph, a `[link](www.google.com)` and an image:

`![Wally](wally.jpg)`

- `* These are`
- `* bullet points`

`> "To be, or not to be, that is the question."`

`^[*Hamlet*, Act III, Scene I]`

- `1. And this is`
- `1. a numbered`
- `7. list`

Markdown output

Here is some text in *italics*, in **bold** and teletype.

Here is a new paragraph, a link and an image:



- These are
- bullet points

*“To be, or not to be, that is the question.”*¹

- ① And this is
- ② a numbered
- ③ list

¹*Hamlet*, Act III, Scene I

Markdown tables

```
| Left | Centre | Right |
| ----- |:-----: | -----: |
| You can | This text is | 42 |
| use Markdown | centre-aligned | 314 |
| within tables | | 37 |
```

Output

Left	Centre	Right
You can	This text is	42
use Markdown	centre-aligned	314
<i>within</i> tables		37

Markdown code chunks

To investigate the relationship between `height` and `weight` we fitted a *simple linear regression model*, as follows.

```
```r
model <- lm(weight ~ height, data = women)
summary(model)
plot(model) # Residual diagnostics
```
```

Output

To investigate the relationship between height and weight, we fitted a *simple linear regression model*, as follows.

```
model <- lm(weight ~ height, data = women)
summary(model)
plot(model) # Residual diagnostics
```

YAML headers

```
---  
title: "The name of my Markdown document"  
author: "David A. Selby"  
date: "27 September 2016"  
output: pdf_document  
---  
  
(content)
```

YAML (yet another markup language) headers let you specify additional options before rendering your document

Markdown: so what?

So far, Markdown is just a lightweight typesetting program.

How will this help you become more productive?

Introducing **R Markdown**...

R Markdown

R Markdown

An ordinary Markdown code chunk:

```
```r  
your R code goes here
```
```

An R Markdown **R code** chunk:

```
```{r}  
your R code goes here
```
```

R Markdown

You can run R **in-line** with text as well. To add in-line R code, we use the syntax `'r your_code_here'`. This will **evaluate and return the result** within the paragraph. For example:

```
If we multiply 13 and 56 we get `r 13 * 56`.  
The date today is `r format(Sys.Date(), "%d %B %Y")`.  
There are `r nrow(iris)` observations in the iris data set.
```

Output

If we multiply 13 and 56 we get 728.
The date today is 16 November 2016.
There are 150 observations in the iris data set.

I heard you like code chunks...



Re-using code chunks

Yo dawg, check out this **cool** plot:

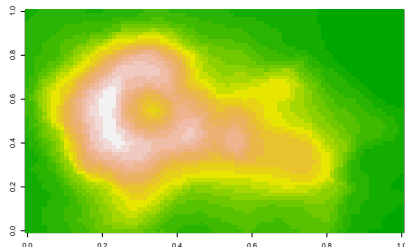
```
```{r chunk1, echo = FALSE}  
image(volcano, col = terrain.colors(20), labels = NULL)
```
```

Here is the code we used to make it!

```
```{r chunk2}  
```{r chunk1, eval = FALSE}  
```
```

## Re-using code chunks (output)

Yo dawg, check out this *cool* plot:



Here is the code we used to make it!

```
image(volcano, col = terrain.colors(20), labels = NULL)
```

# Other programming languages<sup>2</sup>

## A Python code chunk

```
```{python}
x = ['To', 'be', 'or', 'not', 'to', 'be']
y = [i.upper() for i in x]
print(" ".join(y) + 5 * '?!')
```
```

## Output

```
x = ['To', 'be', 'or', 'not', 'to', 'be']
y = [i.upper() for i in x]
print(" ".join(y) + 5 * '?!')
```

```
TO BE OR NOT TO BE?!?!?!?!?!
```

---

<sup>2</sup>Assuming they are installed and on your PATH

# Lazy, productive research

# Nobody need ever know!

- `knitr::kable` or `xtable::xtable` to auto-generate tables
- `echo = FALSE` to hide code in output
- `cache = TRUE` to save results that take a long time to run
- `output: word_document` to generate `.docx` files
- Set a bibliography in YAML, then cite:  
e.g. “As found by `[@fisher1931]`...”

## Another thing R Markdown is great for

*Will finish this slide later...*

# Outreach in R Markdown

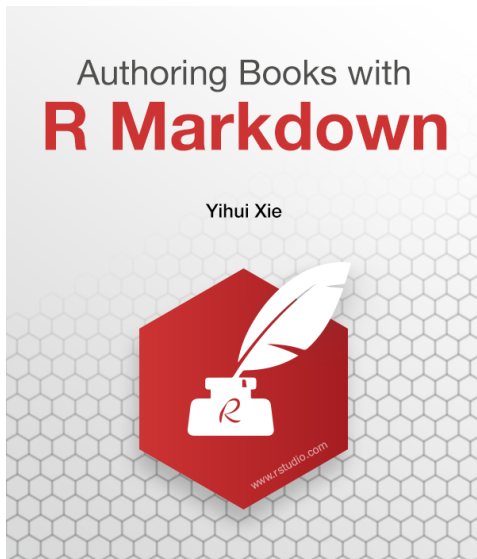
**Jekyll** transforms Markdown into static websites and blogs

**GitHub Pages** serves and hosts Jekyll web sites for free

`knitr-jekyll` Automatically knits R Markdown documents, builds them with Jekyll and serves them locally



# Write your entire thesis in R Markdown



<http://www.bookdown.org>

# Links & further reading

**Literate Programming** Donald Knuth (1992)

**R Markdown** <http://rmarkdown.rstudio.com>

**knitr** <http://yihui.name/knitr>

**R Markdown reference guide and cheat sheet**

<https://www.rstudio.com/resources/cheatsheets/>

**Advanced R** Hadley Wickham (2014) <http://adv-r.had.co.nz/>

**R packages** Hadley Wickham (2015) <http://r-pkgs.had.co.nz/>