Getting started with R Markdown

Justin Sulik

`http://github.com/justinsulik/` (check the README in the rMarkdown folder for a display issue)

`http://rpubs.com/justinsulik/`

March 17, 2017

Aims

- To show you why R Markdown is useful
  - Basically: automatic creation of reports with text and stats together
- To get you started
- To encourage you to go learn how to do it yourself
  - Not a hands-on tutorial
- Not an R tutorial
  - But maybe motivate you to learn R

Caveat

- I'm an enthusiastic user of R Markdown (not an expert!)
- I've just been using it about 6 months
- I use R Studio on a Mac
  - R studio includes installations of `rmarkdown` and `pandoc`
  - If you're using something else, you might need to install these

What is R Markdown?

- When you start using R, you probably focus on typing commands in the console
- You then move onto scripts
    - Write R commands, run, get output
    - Limited shareability
    - Still have to transfer output into your report
    - Plots and text in separate windows
    - Can leave comments
        - But limited formatting/functionality
    - Any changes require multiple steps
- The next step towards richer, more informative, more efficient stats output is R Markdown

What is R Markdown?

- Not a special new thing
- Just means the document contains both
  - R code
  - Markdown
    - plain-text formatting syntax
    - lightweight
    - easy to read
    - easily converted into html and other formats
    - see examples at the Wikipedia page

What is R Markdown for?

- Everything in one document:
  - R code
  - Your text/comments/discussion
  - Plots
  - Citations
- Highly formattable
  - Rich text markup
  - Pretty tables
  - Headings, navigation aids
  - Bibliography
  - Equations
- With one click, this is transformed into html/pdf/doc/slideshow
- With one more click, can be shared via RPubs `http://rpubs.com/justinsulik/`

Basic structure of an R Markdown document

```
---
title: "Example"
author: "Justin Sulik"
date: "March 15, 2017"
output: html_document
---

This will appear as formattable text where you can describe what you're doing, comment on results, etc.
It's formatted with plain text symbols, yielding *italics*, **bold**, math: $x^2, \sqrt{y}, \sum_i^k m_i$,
headings:

## This is a heading

### This is a subheading

The following is an R chunk

```{r distributions}

myFunction <- function(x){
  return(x+x^2)
}

myFunction(3)

mean(iris$Sepal.Length)

model1 <- lm(Sepal.Length ~ Species, data=iris)

summary(model1)
```

You can also include inline R code too: the mean of Sepal.Length is `r mean(iris$Sepal.Length)`
```

Basic structure of an R Markdown document

YAML header

```
---
title: "Example"
author: "Justin Sulik"
date: "March 15, 2017"
output: html_document
---

This will appear as formattable text where you can describe what you're doing, comment on results, etc.
It's formatted with plain text symbols, yielding *italics*, **bold**, math: $x^2, \sqrt{y}, \sum_i^k m_i$,
headings:

## This is a heading

### This is a subheading

The following is an R chunk

```{r distributions}

myFunction <- function(x){
  return(x+x^2)
}

myFunction(3)

mean(iris$Sepal.Length)

model1 <- lm(Sepal.Length ~ Species, data=iris)

summary(model1)
```

You can also include inline R code too: the mean of Sepal.Length is `r mean(iris$Sepal.Length)`
```

Basic structure of an R Markdown document

```
---
title: "Example"
author: "Justin Sulik"
date: "March 15, 2017"
output: html_document
---

This will appear as formattable text where you can describe what you're doing, comment on results, etc.
It's formatted with plain text symbols, yielding *italics*, **bold**, math: $x^2, \sqrt{y}, \sum_i^k m_i$,
headings:

## This is a heading

### This is a subheading

The following is an R chunk

```{r distributions}

myFunction <- function(x){
  return(x+x^2)
}

myFunction(3)

mean(iris$Sepal.Length)

model1 <- lm(Sepal.Length ~ Species, data=iris)

summary(model1)
```

You can also include inline R code too: the mean of Sepal.Length is `r mean(iris$Sepal.Length)`
```

Markdown

Basic structure of an R Markdown document

```
---
title: "Example"
author: "Justin Sulik"
date: "March 15, 2017"
output: html_document
---

This will appear as formattable text where you can describe what you're doing, comment on results, etc.
It's formatted with plain text symbols, yielding *italics*, **bold**, math: $x^2, \sqrt{y}, \sum_i^k m_i$,
headings:

## This is a heading

### This is a subheading

The following is an R chunk

```{r distributions}

myFunction <- function(x){
  return(x+x^2)
}

myFunction(3)

mean(iris$Sepal.Length)

model1 <- lm(Sepal.Length ~ Species, data=iris)

summary(model1)
```

You can also include inline R code too: the mean of Sepal.Length is `r mean(iris$Sepal.Length)`
```

R code

Sample output

Example

Justin Sulik

March 15, 2017

This will appear as formattable text where you can describe what you're doing, comment on results, etc. It's formatted with plain text symbols, yielding *italics*, **bold**, math: x^2, $\sqrt{y}$, $\sum_i^k m_i$, headings:

This is a heading

This is a subheading

The following is an R chunk

```
myFunction <- function(x){
  return(x+x^2)
}

myFunction(3)
```

```
## [1] 12
```

```
mean(iris$Sepal.Length)
```

```
## [1] 5.843333
```

```
model1 <- lm(Sepal.Length ~ Species, data=iris)

summary(model1)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Species, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6880 -0.3285 -0.0060  0.3120  1.3120
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.0060     0.0728  68.762  < 2e-16 ***
## Speciesversicolor   0.9300     0.1030   9.033 8.77e-16 ***
## Speciesvirginica    1.5820     0.1030  15.366  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5148 on 147 degrees of freedom
## Multiple R-squared:  0.6187, Adjusted R-squared:  0.6135
## F-statistic: 119.3 on 2 and 147 DF,  p-value: < 2.2e-16
```

You can also include inline R code too: the mean of Sepal.Length is 5.8433333

Why is it worth learning?

- Open, reproducible science
- Efficient workflow

Open, reproducible science

- Open data is only half the battle
- Still a large gap between data and conclusions
  - Show your working
  - *Everything* in a paper should be reproducible
- Statistical reporting is error-prone (Nuijten et al. 2016; Bakker and Wicherts 2011)
  - Hidden decisions
  - Copy+paste errors
  - Rounding errors
  - Typos
  - Failure to update results
  - Problems with model but still a readable output
- Not everything can be published

How can R Markdown help?

- It can show your working
  - Precisely what analyses you ran, e.g.
    - if one-sided test used
    - if data centred for interaction term
    - random effects structure
    - what issues cropped up
  - How you ran them
  - No page pressure!
- Avoid silly errors
- No need to do anything manually, or copy+paste

How can R Markdown help?

- Share with whoever
  - Send html link to collaborators
  - Publish with one click
  - Create pdf or html as supplementary material for article
  - Blog
  - Help avoid file-drawer problem

Efficient workflow

- Usually: do the stats in R, write the report (e.g. in Word), put the stats in the Word doc
  - Non-automatic (manually type or copy+paste)
  - Export graphs
  - Lots to keep track of/switching back and forth
  - Plenty of room for error
  - Ugly
- Common time-wasters
  - Get new data, have to include it in the analysis
  - Trying to work out just how you did something 6 months ago
    - Worse: getting a different p-value!
  - One change means you have to re-run a whole bunch of scripts
  - Or spend ages messing with images in Word
  - Lengthy processes (e.g. bootstrapping) painful to repeat
  - Copy+pasting model output messes up columns

Simple choice

Quick, automatic, process with less room for human error

vs.

Slow, error-prone, non-automated process

It's 2017. 'Let the computer do the work' (Wilson et al. 2014)

Getting started

- R Studio has most of what you need already installed
- Might need:
  - `install.packages("knitr")`
  - `library(knitr)`
- While you're at it:
  - install `dplyr, tidyr`

Getting started

- File > New file > R Markdown . . .
- What sort of document do you want to produce?
  - html, pdf, doc, slide show
- Title:
  - Not a file name
  - What will go at the top of your report (so be descriptive)
  - Can edit later
- Will open a new tab in the viewer
  - Includes plenty of examples to get you started
- Click 'knit' to create chosen format (e.g. html)
  - First time, will ask for file name to save
  - Just `exampleName` (will automatically add `.html` or `.pdf`)

Getting started

- Recall: 3 main parts to `.Rmd` file
  - YAML header
  - Markdown
  - R code
- `knitr` is what renders an `.Rmd` into an `.html` file
  - (Well, `pandoc` does much of the work under the hood)

Markdown

- Using plain text symbols (*, _, $, #) to provide formatting instructions
  - `**this will be in bold**` → **this will be in bold**
  - `This requires some math $x^2$` → This requires some math x^2
  - Different from WYSIWYG (what you see is what you get)
  - Same synax used for github, stackexchange, reddit
  - Same idea (different syntax) for LaTeX or HTML
    - Math syntax `$...$` same as LaTeX though
- Google for cheatsheets and print them out
- Getting used to typing two *'s to make something bold doesn't waste as much time as having to manually put statistics in your Word document and isn't actually more complex than hitting CMD+b or CTRL+B

Markdown

- Main use: presenting stats analyses in nicely formatted text
  - creating the results section of an article
  - creating supplementary analyses to share with article
  - creating report to share with collaborators
  - creating a blog

R code

- Multiline (chunks) or inline (see newly created file for examples)
- Can choose how much detail you want output
  - just results
  - code and results
  - code and results and messages
  - just code
  - run code in background, but display nothing
- Let's switch to an actual example
  - example.Rmd at the github link on page 1
    - Check the README file for how to be sure you see my source code
    - You can view this online, or download and open in any text viewer, but probably best to open in R studio
  - example.html at the rpubs link on page 1

Final word: what to do with LaTeX

- `knitr` is very powerful (with `pandoc`):
  - We've been talking about knitting R + Markdown to html
  - You can also knit R + LaTeX to pdf
  - RStudio > Preferences > R Sweave > weave Rnw files using: knitr
    - This allows use of `knitr` functionality like caching chunks
  - File > New File > R Sweave
    - Saves as `.Rnw` rather than `.Rmd`
- Everything you do in your usual LaTeX editor, you can do in R studio!

A couple things about .Rnw

- R chunks look different: start with << >>= and end with @
- You can create separate .Rnw children, but:
  - Include these in the parent document with `\Sexpr{knit_child('childName.Rnw')}`
  - Don't include a preamble in the child document
  - Instead, use `\Sexpr{set_parent('parentName.Rnw')}` if you want to knit the child on its own

Summary

- All your text, stats and plots in one place
- Create .Rmd file, knit to .html (or other format)
- You're writing the R code to run the analysis anyway, so why not build the report around that?
- Why bother?
  - Reproducible science
  - Efficient workflow
  - Make the computer do all the work!

Online resources

Markdown cheatsheet

A nice tutorial

Another nice tutorial

Sharing research using R Markdown

Preventing statistical reporting errors

Bibliographies and citations

A simple .Rmd file

A more complex .Rmd file

Bibliography

Bakker, Marjan, and Jelte M Wicherts. 2011. "The (Mis) Reporting of Statistical Results in Psychology Journals." *Behavior Research Methods* 43 (3). Springer: 666–78.

Nuijten, Michèle B, Chris HJ Hartgerink, Marcel ALM van Assen, Sacha Epskamp, and Jelte M Wicherts. 2016. "The Prevalence of Statistical Reporting Errors in Psychology (1985–2013)." *Behavior Research Methods* 48 (4). Springer: 1205–26.

Wilson, Greg, DA Aruliah, C Titus Brown, Neil P Chue Hong, Matt Davis, Richard T Guy, Steven HD Haddock, et al. 2014. "Best Practices for Scientific Computing." *PLoS Biol* 12 (1). Public Library of Science: e1001745.