The Magic of Markdown

Ted Laderas (laderast@ohsu.edu)

January 13, 2016

Introduction

- What is Markdown?
    - Simple Intro
- Rmarkdown
    - PDFs
    - Interactive Slides

What is Markdown?

John Gruber (of Daring Fireball) originally created Markdown as a simple replacement for HTML for use in message board posts to allow for better formatting.

The wonderful thing about Markdown is that it doesn't get in your way - you can just get in and edit documents using any text editor. Once you learn it, you find that you can work very fast and not worry about how things look on the page until later.

You can worry about translating the markdown to other formats later, which is handled by an engine called Pandoc.

Markdown Variants

Markdown was originally made to be a substitute for HTML in forums. The original implementation doesn't really cover formatting such as tables.

GitHub also uses their own flavor of Markdown (called GitHub markdown) as the main format for their webpages, which makes maintaining them much easier than having to edit raw HTML. They extended markdown so you can add tables and better code formatting.

A variation of GitHub markdown is Rmarkdown, which is Markdown + R. Rmarkdown is really useful for reproducible analyses, and it can also be used in conjunction with Shiny to make interactive slides.

Markdown Basics

When in doubt, look at this quick markdown cheat sheet that covers both plain markdown and GitHub Markdown.

```
*Italicise your text* _Italicise your text_
```

Italicise your text

```
**Bold your text** __Bold your text__
```

Bold your text

Bulleted Lists

```
* Adding bullets
    * Another depth (use tabs)
```

- Adding bullets
    - Another Depth

Numbered Lists

```
1. Numbered Lists
2. Another Number
```

1. Numbered Lists
2. Another Number

More Markdown (1)

```
Links are automatically generated for URLs:
http://yahoo.com http://yahoo.com
```

```
[Link to Yahoo](http://yahoo.com)
```

Link to Yahoo

Markdown Images

```
![Add an Image](images/giphy.gif)
```

Figure 1: Image

Markdown Code Blocks

Use three backticks “‘ to enclose a code block:

~~~~
```
Put Code Here
Put Code Here
for(i in 1:5){
  print(paste0("I said 'I said' ", 2*i, "times"))
}
```
~~~~

~~~~
Put Code Here
for(i in 1:5){
  print(paste0("I said 'I said' ", 2*i, "times"))
}
~~~~

Even More Markdown

Use a single backtick ‘ to enclose `in-line code`

Use > for blockquotes:

```
> This is a block quote
> continued on the next line
```

> *This is a block quote Continued on the next line*

Escaping Characters

The \ is an escape character and allows you to use *, > and other markdown characters as is.

```
Example: \` \* \#
```

Example: ‘ * #

YAML

YAML* is another way of providing Pandoc the necessary metadata it needs (output format, location of BibTeX library, other executables, etc). For more info, consult the pandoc documentation.

You add YAML as a header to the document by specifiying three dashes: `---`

Here's the YAML that I used for this set of slides.

```
---
title: "The Magic of Markdown"
author: "Ted Laderas"
date: "January 13, 2016"
output: slidy_presentation
---
```

*'YAML ain't markup language' - har har.

Rmarkdown

Rmarkdown is an R-specific version of GitHub markdown (Technically it's based on sundown, but who cares.).

It's used a lot in making analyses and reports reproducible. It allows for Markdown formatting mixed with R analysis. For this reason, it's ideal for sharing complex analyses with other people.

Rmarkdown is translatable to PDF (via LaTeX), to PDF slides, HTML reports, and HTML slides within RStudio.

RStudio has pandoc built in for this purpose.

Rmarkdown Example

The key difference in Rmarkdown is in the codeblock, which actually executes code.

~~~
```{r, eval=TRUE}
data(iris)
plot(iris[,1], iris[,2])
```
~~~

Lots of codeblock options

We've already seen the `eval=TRUE` option for the R codeblock. But there are lots of others:

- `echo=FALSE`
- `fig.size=5`
- `message=FALSE`
- `tidy=TRUE`
- `warning=FALSE`

Rstudio and Pandoc

RStudio actually has pandoc built in, with a limited set of options. If you want your markdown to execute code, you will have to use this version of pandoc. I usually just use Rstudio's "Knit" option to translate my documents.

You can also use the following command to render a document. Make sure your YAML specifies the options.

```
rmarkdown::render("input.Rmd")
```

Rstudio/Markdown Example:

Open this file in RStudio: `http://github.com/laderast/magic-of-markdown/Rmarkdown-example.Rmd`

Shiny and Rmarkdown

What's especially cool is that you can mix Shiny and Rmarkdown to produce interactive slides.

You can embed Shiny applications into code blocks and then run the resulting code on a Shiny server, such as the one available on church.

I'm experimenting with this here: `http://church.ohsu.edu:3838/laderast/clusteringLecture/`

For More Info

- This document: http://github.com/laderast/magic-of-markdown/
- Pandoc user guide: http://pandoc.org/README.html
- Rmarkdown: http://rmarkdown.rstudio.com
- Rmarkdown and Shiny: http://rmarkdown.rstudio.com/authoring_shiny_widgets.html
- GitHub Pages: https://pages.github.com