

R Markdown and reproducible research

Math 251 Calculus 3

September 16, 2013

The idea of markup languages

Word processing software—mostly, Microsoft Word—is so pervasive that most people fire it up without thinking whenever they need to do a bit of writing. For the last 20 years or so, WYSIWYG word processors have dominated when it comes to text documents.

In a word processor, you style individual text elements directly: bold, indented, red, and so on.

Markup languages are different. The document author “marks up” the document with logical structure rather than specific display instructions.

An example of logical markup

```
<title>Example document</title>
<body>
  <h1>Excellent example</h1>
  This document is a toy example. Its markup format
  is <em>modified</em> HTML.
  <ul>
    <li>These</li>
    <li>Are</li>
    <li>List</li>
    <li>Items.</li>
  </ul>
</body>
```

HTML

Most people's direct experience of markup is limited, instead being mediated by web browsers. HTML is the language of the Web, and it stands for **H**ypertext **M**arkup **L**anguage.

HTML consists of text and “HTML elements”, which instruct the browser on the various structural elements of the document: title and author, headings at various levels, paragraphs, block quotes, ordered or unordered lists, emphasized text, links, images, and so on.

HTML does *not* tell the browser how to render the document to the screen. The browser makes these decisions autonomously, often with help from an external style sheet.

Plain text: the way of the Jedi

- ▶ HTML documents are stored as *plain text*

The browsers that render HTML may pass on into oblivion, but the files will remain human-readable.

Plain text: the way of the Jedi

- ▶ HTML documents are stored as *plain text*
 - ▶ Editors, word processors, and file formats come and go

The browsers that render HTML may pass on into oblivion, but the files will remain human-readable.

Plain text: the way of the Jedi

- ▶ HTML documents are stored as *plain text*
 - ▶ Editors, word processors, and file formats come and go
 - ▶ as anyone who has dealt with Word's “compatibility issues” knows

The browsers that render HTML may pass on into oblivion, but the files will remain human-readable.

Plain text: the way of the Jedi

- ▶ HTML documents are stored as *plain text*
 - ▶ Editors, word processors, and file formats come and go
 - ▶ as anyone who has dealt with Word's "compatibility issues" knows
 - ▶ If you want your documents to be readable in 20 years, or even 5 years, plain text is your best bet

The browsers that render HTML may pass on into oblivion, but the files will remain human-readable.

Plain text: the way of the Jedi

- ▶ HTML documents are stored as *plain text*
 - ▶ Editors, word processors, and file formats come and go
 - ▶ as anyone who has dealt with Word's "compatibility issues" knows
 - ▶ If you want your documents to be readable in 20 years, or even 5 years, plain text is your best bet
- ▶ Plain text is also fun to use

The browsers that render HTML may pass on into oblivion, but the files will remain human-readable.

Plain text: the way of the Jedi

- ▶ HTML documents are stored as *plain text*
 - ▶ Editors, word processors, and file formats come and go
 - ▶ as anyone who has dealt with Word's "compatibility issues" knows
 - ▶ If you want your documents to be readable in 20 years, or even 5 years, plain text is your best bet
- ▶ Plain text is also fun to use
 - ▶ The best programs for editing plain text are obscure and powerful, so you will feel like a hacker kung fu wizard

The browsers that render HTML may pass on into oblivion, but the files will remain human-readable.

Plain text: the way of the Jedi

- ▶ HTML documents are stored as *plain text*
 - ▶ Editors, word processors, and file formats come and go
 - ▶ as anyone who has dealt with Word's "compatibility issues" knows
 - ▶ If you want your documents to be readable in 20 years, or even 5 years, plain text is your best bet
- ▶ Plain text is also fun to use
 - ▶ The best programs for editing plain text are obscure and powerful, so you will feel like a hacker kung fu wizard
 - ▶ Plain text will make you seem mysterious and attractive at parties

The browsers that render HTML may pass on into oblivion, but the files will remain human-readable.

Markdown: a plain-text road to HTML

Markdown is another plain text format, designed to be easy to write. It was invented for the blogging community, but has found wide use in academia and scientific circles as well.

The idea is that the simpler HTML elements are easily generated by plain text shorthand. For example, headings are set off by one or more `#` characters (up to six, since HTML provides six levels of subheading). Paragraphs are separated—naturally—by blank lines. Text is *emphasized* by surrounding it with `*` asterisks.

1. Ordered lists are simple to generate.

Markdown: a plain-text road to HTML

Markdown is another plain text format, designed to be easy to write. It was invented for the blogging community, but has found wide use in academia and scientific circles as well.

The idea is that the simpler HTML elements are easily generated by plain text shorthand. For example, headings are set off by one or more `#` characters (up to six, since HTML provides six levels of subheading). Paragraphs are separated—naturally—by blank lines. Text is *emphasized* by surrounding it with `*` asterisks.

1. Ordered lists are simple to generate.
2. So are unordered lists; they can be

Markdown: a plain-text road to HTML

Markdown is another plain text format, designed to be easy to write. It was invented for the blogging community, but has found wide use in academia and scientific circles as well.

The idea is that the simpler HTML elements are easily generated by plain text shorthand. For example, headings are set off by one or more `#` characters (up to six, since HTML provides six levels of subheading). Paragraphs are separated—naturally—by blank lines. Text is *emphasized* by surrounding it with `*` asterisks.

1. Ordered lists are simple to generate.
2. So are unordered lists; they can be
 - ▶ Nested

Markdown: a plain-text road to HTML

Markdown is another plain text format, designed to be easy to write. It was invented for the blogging community, but has found wide use in academia and scientific circles as well.

The idea is that the simpler HTML elements are easily generated by plain text shorthand. For example, headings are set off by one or more `#` characters (up to six, since HTML provides six levels of subheading). Paragraphs are separated—naturally—by blank lines. Text is *emphasized* by surrounding it with `*` asterisks.

1. Ordered lists are simple to generate.
2. So are unordered lists; they can be
 - ▶ Nested
 - ▶ or even mixed, using indentation to indicate nesting.

Markdown is usually parsed into HTML

There are many utilities that can read Markdown source text and generate a fully-fledged web page. All the modules for this course are written in Markdown, and then the utility `pandoc` is used, together with a basic template and some CSS to weave them into HTML files.

In fact, the workshops and slide decks are also written in Markdown. Using different templates and `pandoc` commands, they are passed to the sophisticated text rendering engine LaTeX to produce the handouts and slide shows.

You don't need any of this fancy stuff to make good use of Markdown. There is extensive R support for Markdown built into R Studio.

R Markdown and scientific computing with reporting

The literate programming tradition has necessitated the development of tools to combine visually pleasing text with source code. In the R world, the tool of choice for many years was Sweave (S + “weave”, for weaving the code and text together).

Sweave allowed computations to be performed *in the actual reporting document* rather than in a separate environment. However, to use Sweave, you must know how to write documents in LaTeX, which is beyond the scope of this course.

A modern, lightweight alternative to Sweave is `knitr`, a package available from within R Studio.

What is knitr?

The `knitr` package allows you to use Markdown and R code in the same document, which is then easily (seriously: one click) rendered into HTML. Any R computations or plots you generate are included into the final HTML document.

By default, the output is like a standard web page, but with the help of templates, CSS style information, and powerful text processing engines such as `pandoc`, any kind of HTML document can be generated. In particular, it is possible to create impressive HTML5 slideshows containing professionally typeset mathematics and R-generated images without surprisingly little effort.