SOC 5050: Markdown Quick Reference

Christopher Prener, Ph.D.

August 29th, 2016

What is Markdown?

Markdown is a simple markup language. Markup languages are used to give computer programs directions on how particular blocks of text should be processed. Markdown was developed in 2004 by writer and developer John Gruber. Gruber describes Markdown on his website:

Markdown is intended to be as easy-to-read and easy-to-write as is feasible.

Readability, however, is emphasized above all else. A Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. While Markdown's syntax has been influenced by several existing text-to-HTML filters — including Setext, atx, Textile, reStructuredText, Grutatext, and EtText — the single biggest source of inspiration for Markdown's syntax is the format of plain text email.

To this end, Markdown's syntax is comprised entirely of punctuation characters, which punctuation characters have been carefully chosen so as to look like what they mean. E.g., asterisks around a word actually look like emphasis. *Markdown lists look like, well, lists. Even blockquotes look like quoted passages of text, assuming you've ever used email. . .*

. . . Markdown's syntax is intended for one purpose: to be used as a format for writing for the web.

Markdown Syntax

As markup syntaxes go, Markdown is exceptionally straight forward. The following sections include examples of syntax used to create Markdown documents.[1] These are specific to what is called GitHub Markdown - there are some subtle differences in the way GitHub uses Markdown formatting.

[1] If you want to see how these examples render in Markdown, a matching `.md` file is available on GitHub that you can preview.

Headings   Markdown contains six heading levels. Headings are identified with # symbols:

```
# This is the largest heading
## This is the second largest heading
###### This is the smallest heading
```

New Paragraphs   Leave a blank line between two lines of text to create a new paragraph.

Styling Text   Text can be styled using bold, italics, and strikethroughs. To create italicized text, wrap your text with a single asterisk `*`. To create bold text, wrap your text with double asterisks `**`. To create strikethrough text, wrap your text with two tildes `~~ ~~`.

```
*This is an italicized sentence.*

**This is a bolded sentence.**

~~This is a sentence with strikethrough text~~
```

Quoting Text   Quoting text (which I have used above to illustrate examples) is done with a greater then symbol `>`.

Quoting Code   There are two types of code quotes in Markdown. Inline quotes, which are included in a sentence, are wrapped in single backticks:

```
Use the 'describe' command to list variables in Stata.
```

To include code blocks, which are better for including the full syntax of particular commands and their output, use triple backticks:

```
'''Stata
. describe make price mpg

            storage   display    value
variable name   type    format     label       variable label
-----------------------------------------------------------------------------
make            str18   %-18s                  Make and Model
price           int     %8.0gc                 Price
mpg             int     %8.0g                  Mileage (mpg)
'''
```

Note how the word 'Stata' is written after the first set of triple backticks. This is an indicator for GitHub that the code is written in

Stata's programming language. By including this, GitHub can apply some syntax highlighting to your files. This makes them easier to read.

Links   In Markdown, adding hyperlinks is a two step process. The text that you want to have hyperlinked is written first and is wrapped in brackets `[]`. After this, you include the URL wrapped in parentheses `()`. This is an example of including in-line hyperlinks:

```
The course [website](https://github.com/slu-soc5050) is hosted using the service
[GitHub](https://github.com).
```

Embedding Images   Within the directory that contains your Markdown file, create a subdirectory called 'Images'. Save all images for a particular assignment there. In your main assignment Markdown file, include a hyperlink reference:

```
![screenshot1](https://github.com/slu-soc5050/jdoe/blob/master/ps1/images/image1.png)
```

Note how, instead of including text to be hyperlinked, we suppress this aspect of the syntax by using an exclamation point `!`.

Simple Lists   Bulleted lists are indicated in Markdown using the dash `-` or a single asterisk `*`:

```
- mean
- median
- mode
* variance
* standard deviation
```

Enumerated lists are created by preceding each line with the appropriate number:

```
1. calculate the mean
2. calculate the variance
3. calculate the standard deviation
```

You can create more complex lists by preceding a line with two single spaces. You can also combine bulleted and enumerated lists when using this approach.

Task Lists   If you want to create task lists on GitHub, you can include brackets separated by a space before each list item `[]`. Completed tasks include an `x` in place of the space `[x]`. These task lists are interactive - when published on GitHub, you can click on the resulting checkboxes to toggle them between complete / incomplete.

```
1. [x] calculate the mean
2. [ ] calculate the variance
3. [ ] calculate the standard deviation
```

Mentioning Other GitHub Users   If you want to mention me or one of your classmates in a comment, include the `@` symbol before their username:

```
Hey @chris-prener, thanks for the feedback. I made the changes to lines 40 and 41.
```

Once the document is uploaded to GitHub, the username will render as a hyperlink and the user will be alerted.

Previewing Markdown   If you write your Markdown documents in Atom, you can use the Markdown Preview package to generate an interactive preview of your document. As you type, the preview will update. To open this preview in a tab, go to Packages ▷ Markdown Preview ▷ Toggle Preview in Atom.

Document Details

Document produced by Christopher Prener, Ph.D. for the Saint Louis University course SOC 5050 - QUANTITATIVE ANALYSIS: APPLIED INFERENTIAL STATISTICS. See the course wiki and the repository README.md file for additional details.