

An Example Github Pages Document

June 2013

Here is some text about the document - something like an abstract ... Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur lobortis lectus et tellus pharetra sodales. Duis eget sapien quis urna sagittis facilisis. Ut aliquam dui ut est euismod a mattis magna semper. Curabitur nec magna turpis. Etiam suscipit lectus vel lacus dignissim sollicitudin non at nisi. Proin ut nibh non magna vulputate laoreet in eget felis. Vivamus eget turpis nulla. Aliquam erat volutpat. Suspendisse potenti. Nullam viverra diam sit amet risus fermentum iaculis. Integer vitae purus eu urna ultrices sodales vel sed nisl. Cras lorem est, ultrices sed vulputate in, mattis ac justo.

1 Introduction

This Github repository provides a template for transforming a simple markdown text file into a nice looking HTML document by using Jekyll. Because Github supports Jekyll, this document can be hosted as a static website for free through [Github pages](#). You are now looking at the sample version of this document, which includes text (including lists, sections, subsections and sub-subsections), as well as tables and figures of varying levels of sophistication. Some examples use HTML code integrated directly into the markdown text file, but when possible I have provided simple markdown analogues. As would be expected, a more detailed document requires more detailed code ...

1.1 Lists and YAML Front Matter

This section provides an example of how to form basic lists and also provides an explanation of the YAML front matter required to produce this document. What in the world is YAML front matter, you ask? [Read this](#) for a general information and [this article](#) for an explanation of how Jekyll (and Liquid) use it to produce static websites. YAML front matter is located at the very top of the `index.md` file. It is composed of several variables sandwiched between triple dashes (`---`). These variables define how your Markdown Document will appear and will require some customization by you the author.

1. **"layout"** - this corresponds to the *.html file you want to use as the 'layout' for your document. Two choices are provided here in the `_layouts/` directory. The first is `report.html`, which is the more detailed document that includes the "document lens" navigation tool on the left hand side of the static website. The other layout template is `report_simple.html`, which provides only the formatted document. It allows one to print out a clean version of the document, and may be more easily read on mobile devices. This version [can be viewed here](#), although a link for it can also be found at the bottom of the document lens. The markdown file used to create it is identical to `index.md`, but with the revised YAML entry for 'layout' and has the 'contents' area removed from it. The raw version of the markdown text file can be [viewed on Github](#).
 2. **"report_title"** - This is the printed title at the top of the document. It is also the name that must be given (words separated by underscores) to both the pdf version of the report for download and the .md file used for the simple (printable) version. Links for these items are created as a default.
 3. **"report_abstract"** - this is the blurb that (can) appear underneath the document title.
 4. **"report_date"** - the date the report/document was published
 5. **"meta_title"** - The informational title that appears on the web-browser tab that the page is displayed in.
 6. **"meta_subtitle"** - The informational subtitle that appears on the web-browser tab. Separated from the meta title by a colon (:).
 7. **"contents"** - These are the document content descriptors that create the "document lens" navigation tool on the left hand side of the static website. This is a compound YAML entry that is comprised of two components:
 1. **"sections"** - the 'header 1' in markdown and/or 'h1' in html (sections beginning with single # marks)
 2. **"subsections"** - the 'header 2' in markdown and/or 'h2' in html (double # marks)
- **NOTE.** The sections and subsections provided in the **"contents"** must be IDENTICAL to the actual section names in order to produce the link to the (sub)section. However, one can provide an alternate name in the document lens with the

use of direct HTML coding. See the following section [Long Section Names](#) for more details:

- **NOTE.** Although HTML has up to 6 headers (h6), only the first 2 are currently supported in the “Document Lens”
- **NOTE.** Did you notice that the **Notes** are a totally separate list than the numbered one above? GROOVY...

1.2 Long section names can be shortened in the document navigation lens ...

Shortened header names can be included in the document navigation lens (left side of page) by using HTML id code added to the header line. For example, this h2 subsection is referred to as simply “Long Section Names” in both the YAML front matter and the document navigation. The header line is given in the markdown document as

```
## Long section names can be shortened in the document navigation lens ... <a id="long-section-names">.
```

1.2.1 Sub-Sub Sections

Third level sections get included in the section numbering system (here 1.2.1), but the DO NOT get included in the “Document Lens” navigation.

1.2.2 Curabitur sodales ligula in libero

Sed dignissim lacinia nunc. Curabitur tortor. Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis tristique sem. Proin ut ligula vel nunc egestas porttitor. Morbi lectus risus, iaculis vel, suscipit quis, luctus non, massa. Fusce ac turpis quis ligula lacinia aliquet. Mauris ipsum. Nulla metus metus, ullamcorper vel, tincidunt sed, euismod in, nibh. Quisque volutpat condimentum velit.

2 Equations and Tables

This document includes some pretty fine features, including the ability to render math equations, tables and figures. I use the kramdown flavor of markdown to allow for LaTeX style math formulas (with the use of MathJax). LaTeX tables can also be rendered if one treats an ‘array’ as a math equation. Unfortunately some of the nice features available in the tabular environment are not available in array (e.g. the `\multicolumn` feature).

There are three types of table examples provided in this section. The degree of sophistication and difficulty increases from basic markdown tables to LaTeX and bona-fide HTML script included directly within the markdown text document.

2.1 Mathematical Equations

When “kramdown” flavor of Markdown is specified in the `_config.yml` file, it is possible to produce beautiful equations, either inline or as a separate “display” line. Equations are written in LaTeX, interpreted into HTML by MathJax. For example, to include an inline LaTeX equation you enclose the code in `$` delimiters. For example, this code `$\frac{1}{n} \sum_{i=1}^n x_i$` for the arithmetic mean produces this $\frac{1}{n} \sum_{i=1}^n x_i$ inline equation. Other, more complex or important, equations may need their own line, or “display”. Displayed equations are also numbered according to the order of presentation. For example:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{\theta}_i - \theta_i)^2 \quad (1)$$

$$MSE(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \theta)^2] \quad (2)$$

2.2 Markdown Tables

The nicest example tables are provided using raw HTML in subsequent sections. This, however, sucks and kinda defeats the purpose of using markdown to begin with. This section provides a simple example of a table produced in ‘pure’ markdown. None of these options seem great to me, so if you have a better (i.e. ‘pure’ markdown) solution please let me know by forking this repository and placing a pull request, or by adding an [issue](#).

2.2.1 Basic Markdown Tables

There are many simple examples of markdown tables. A cheatsheet for this and other good stuff can be [found on this page](#). Here are their examples:

Colons can be used to align columns.

Tables	Are	Cool
col 3 is	right-aligned	\$1600
col 2 is	centered	\$12
zebra stripes	are neat	\$1

The outer pipes (`|`) are optional, and you don't need to make the raw Markdown line up prettily. You can also use inline Markdown.

Markdown	Less	Pretty
<code>Still</code>	<code>renders</code>	<code>nicely</code>
1	2	3

2.2.2 Complex Markdown Tables

Here is an attempt I made at making a more complex markdown table using seperators to mark two main sections of the table (**Aliquam Leo Lorem** and **Aliquam Leo Nibh**). Multiple column spanning is not possible, so I try to place and align columns as best as possible.

Table 1.

	Aliquam Leo Lorem			Aliquam Leo Nibh		
	Nec Nibh	Ultricies	Cras Purus	Nec Nibh	Ultricies	Cras Purus
Pellentesque	362.980	0.019	False	393.244	-0.107	True
Pellentesque Nec	360.795	-0.001	True	379.277	0.088	False

Fusce ac turpis quis ligula lacinia aliquet. Mauris ipsum. Nulla metus metus, ullamcorper vel, tincidunt sed, euismod in, nibh. Quisque volutpat condimentum velit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam nec ante. Sed lacinia, urna non tincidunt mattis, tortor neque adipiscing diam, a cursus ipsum ante quis turpis. Nulla facilisi. Ut fringilla. Suspendisse potenti. Nunc feugiat mi a tellus consequat imperdiet. Vestibulum sapien.

2.3 LaTeX Tables

Tables can also be produced using the LaTeX “array” environment. An array can be rendered as a table when presented as math code. Although array is a math mode analog to the tabular environment, some of the functionality of tabular is unavailable in array. For example, Table 3 would require the use of ‘\multicolumn’ to do the job of HTML codes ‘colspan’. Here is some [documentation about LaTeX arrays](#).

Like [display LaTeX math functions](#), the array is enclosed by double dollar sign, `$$`, in the markdown text. Here is an example of a simple LaTeX array used in markdown.

Table 4. Disply Math Function (Centered, but puts equation number in)

What	Variable	Integer	
First number	x	8	
Second number	y	15	
Sum	$x + y$	23	
Difference	$x - y$	-7	
Product	xy	120	(3)

[Inline LaTeX](#) can also be used. Here the array is enclosed in single `$` in the markdown text.

Table 5. Inline Math Function (Aligned Left, requires quadruple backslashes for ‘end line’)

What	Variable	Integer
First Number	x	8
Second Number	y	15
Sum	$x + y$	23
Difference	$x - y$	-7
Product	xy	120

Etiam ultrices. Suspendisse in justo eu magna luctus suscipit. Sed lectus. Integer euismod lacus luctus magna. Quisque cursus, metus vitae pharetra auctor, sem massa mattis sem, at interdum magna augue eget diam. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi lacinia molestie dui. Praesent blandit dolor. Sed non quam. In vel mi sit amet augue congue elementum. Morbi in ipsum sit amet pede facilisis laoreet.

2.4 HTML Tables

As I have mentioned, HTML tables are potentially more flexible, beautiful and awesome than either LaTeX or Markdown tables. But they make me want to cry when I see the code and/or try to interpret. I wish they were easier to produce and read, but they are not. It may defeat the purpose of using markdown to begin with slightly, but at least the option exists.

2.4.1 First HTML Example Table

Table 2. A Table Produced Using HTML Code Directly within the Document

	Aliquam Leo Lorem			Aliquam Leo Nibh		
	Nec Nibh	Ultricies	Cras Purus	Nec Nibh	Ultricies	Cras Purus
Analysis						
Pellentesque	362.98	0.02	8.00	287.78	-0.428	5.65
Pellentesque Nec	360.80	-0.00	7.51	379.28	-0.57	6.23

2.4.2 Second HTML Example Table

HTML Provides the most flexible table presentation I've found. Here is another example

Table 3. Another Table Produced Using HTML Directly

	Mean Amet egestas			Median Amet egestas		
	Nec Nibh	Ultricies	Cras Purus	Nec Nibh	Ultricies	Cras Purus
Pellentesque						
Without Dictumst	5.57	-0.20	2.36	26.37	-0.10	5.14
With Dictumst	4.74	-0.08	2.18	28.36	0.03	5.33
Pellentesque Nec						
Without Dictumst	5.95	-0.26	2.43	27.10	-0.27	5.20
With Dictumst	4.52	0.09	2.13	25.68	0.08	5.06

3 Figures and Charts

Here are some examples of how to include figures and charts. These examples range from simple static images stored in a local directory or sourced from the web to interactive [D3 charts] produced using the [rCharts](#) package for the [R statistical programming language](#).

3.1 Static Figures

Figure 1. From a File in the Local Directory



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

3.2 Images from the Web



Fig. 1 Although this is the second figure of this document, it is labeled as 'Fig. 1' because it uses HTML code that is recognized as a specific 'class' in the document that allows for automatic figure counting (controlled in document_print.css).

3.3 Interactive D3 Charts

Static charts and plots can be saved as .png, .jpg or other raw files and included in a similar fashion to the two methods shown above. Here I show how more interesting, interactive figures can be included. Specifically, the following 3 charts have been produced using the `rCharts` package available for the open-source R statistical programming language. This R package leverages the `D3 JavaScript library` on the backend. D3 stands for 'Data-Driven Documents'.

3.3.1 D3 Barchart example using a .js script file

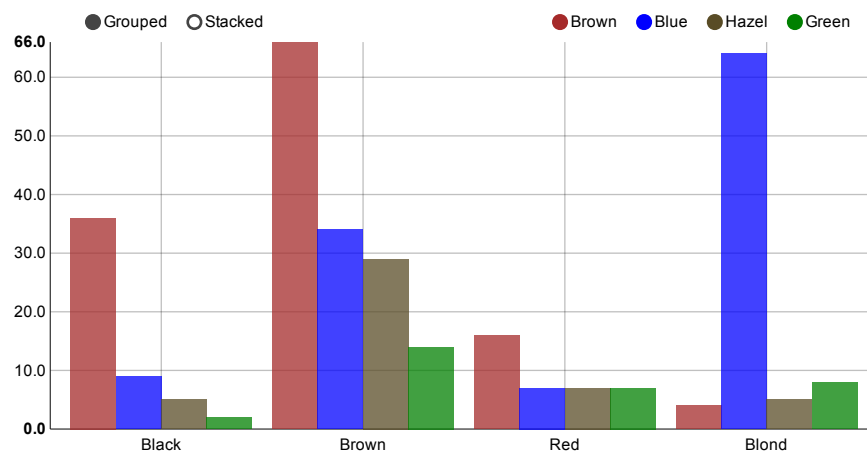


Fig. 2 Example interactive bar chart created using the `rChart` package in R. Click on the circles at the top of the chart to customize the display.

3.3.2 D3 Scatterplot example using inline HTML 'script'

The 2nd chart was created using these commands:

```
p1 <- nPlot(mpg ~ wt, group = 'cyl', data = mtcars, type = 'scatterChart')
p1$xAxis(axisLabel = 'Weight')
p1
```

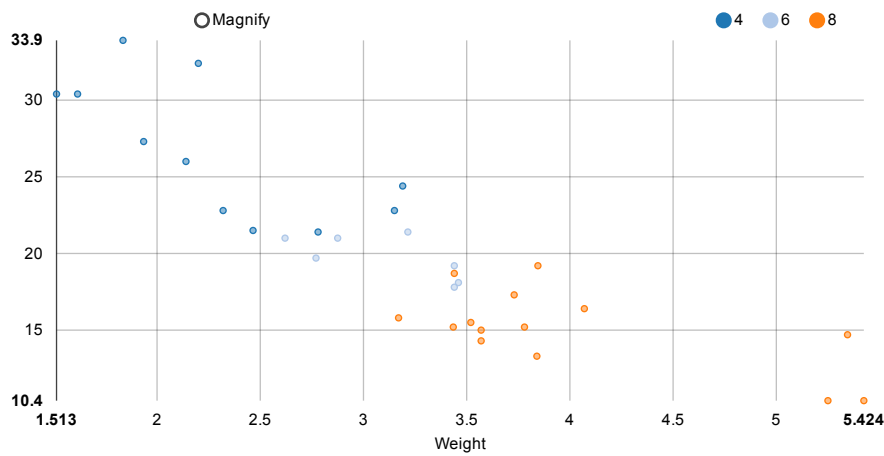


Fig. 3 Example interactive scatter chart created using the *rChart* package in R. Click on the circles at the top of the chart to customize the display.

D3 plots can also be saved as a stand alone HTML file and linked to from the main document, as shown here. These .html files must be saved in the base directory (rather than the `img/` directory for example) so that they have access to assets such as the .js and .css scripts.

3.3.3 A third D3 *rChart* example, because I just can't get enough ...

This 3rd chart was created using these commands:

```
dat <- data.frame(t=rep(0:23,each=4),var=rep(LETTERS[1:4],4),val=round(runif(4*24,0,50)))
p8 <- nPlot(val ~ t, group = 'var', data = dat, type = 'stackedAreaChart', id = 'chart')
p8
```

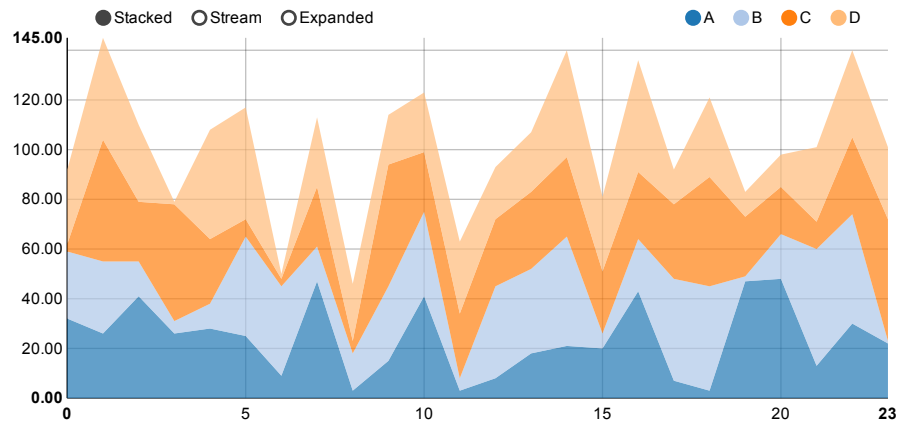


Fig. 4 Example interactive Stacked Area Chart created using the *rChart* package in R. Click on the circles at the top of the chart to customize the display.

4 References

Doe, J. (2010). *First Book*. Cambridge: Cambridge University Press.

Doe, J. (2012). Article. *Journal of Generic Studies*, 6, 33–34.

Doe, J., Smith, S., & Roe, J. (2007). Why Water Is Wet: A really long citation entry to provide an example of the hanging indent used in this document's citation div class. In S. Smith (Ed.), *Third Book*. Oxford: Oxford University Press.

