CSV Tables in Markdown — Pandoc Filter for CSV Tables

November 30, 2016

Contents

The pantable package comes with 2 pandoc filters, `pantable.py` and `pantable2csv.py`. `pantable` is the main filter, introducing a syntax to include CSV table in markdown source. `pantable2csv` complements `pantable`, is the inverse of `pantable`, which convert native pandoc tables into the CSV table format defined by `pantable`.

Some example uses are:

1. You already have tables in CSV format.
2. You feel that directly editing markdown table is troublesome. You want a spreadsheet interface to edit, but want to convert it to native pandoc table for higher readability. And this process might go back and forth.
3. You want lower-level control on the table and column widths.
4. You want to use all table features supported by the pandoc's internal AST table format, which is not possible in markdown for pandoc <= 1.18.[1]

1 `pantable`

This allows CSV tables, optionally containing markdown syntax (disabled by default), to be put in markdown as a fenced code blocks.

[1] In pandoc 1.19, grid-tables is improved to support all features available to the AST too.

1.1 Example

Also see the README in GitHub Pages. There's a LaTeX output too.

````
```table
---
caption: '*Awesome* **Markdown** Table'
alignment: RC
table-width: 0.7
markdown: True
---
First row,defaulted to be header row,can be disabled
1,cell can contain **markdown**,"It can be aribrary block element:

- following standard markdown syntax
- like this"
2,"Any markdown syntax, e.g.",$$E = mc^2$$
```
````

becomes

Table 1: *Awesome* **Markdown** Table

First row	defaulted to be header row	can be disabled
1	cell can contain **markdown**	It can be aribrary block element: • following standard markdown syntax • like this
2	Any markdown syntax, e.g.	$E = mc^2$

(The equation might not work if you view this on PyPI.)

1.2 Install and Use

Install:

```
pip install -U pantable
```

Use:

```
pandoc -F pantable -o README.html README.md
```

1.3 Syntax

Fenced code blocks is used, with a class `table`. See Example.

Optionally, YAML metadata block can be used within the fenced code block, following standard pandoc YAML metadata block syntax. 7 metadata keys are recognized:

- `caption`: the caption of the table. If omitted, no caption will be inserted. Default: disabled.
- `alignment`: a string of characters among `L,R,C,D`, case-insensitive, corresponds to Left-aligned, Right-aligned, Center-aligned, Default-aligned respectively. e.g. `LCRD` for a table with 4 columns. Default: `DDD...`
- `width`: a list of relative width corresponding to the width of each columns. e.g.

  ```
  - width
      - 0.1
      - 0.2
      - 0.3
      - 0.4
  ```

  Default: auto calculated from the length of each line in table cells.
- `table-width`: the relative width of the table (e.g. relative to `\linewidth`). default: 1.0
- `header`: If it has a header row or not. True/False/yes/NO are accepted, case-insensitive. default: True
- `markdown`: If CSV table cell contains markdown syntax or not. Same as above. Default: False
- `include`: the path to an CSV file, can be relative/absolute. If non-empty, override the CSV in the CodeBlock. default: None

When the metadata keys is invalid, the default will be used instead.

1.4 Related Filters

The followings are pandoc filters written in Haskell that provide similar functionality. This filter is born after testing with theirs.

- baig/pandoc-csv2table: A Pandoc filter that renders CSV as Pandoc Markdown Tables.
- mb21/pandoc-placetable: Pandoc filter to include CSV data (from file or URL)
- sergiocorreia/panflute/csv-tables.py

	pandoc-csv2table	pandoc-placetable	panflute example	pantable
caption aligns	caption aligns = LRCD	caption aligns = LRCD	title	caption aligns = LRCD

	pandoc-csv2table	pandoc-placetable	panflute example	pantable
width		widths = "0.5 0.2 0.3"		width: [0.5, 0.2, 0.3]
table-width				table-width: 1.0
header	header = yes | no	header = yes | no	header: True | False	header: True | False | yes | NO
markdown		inlinemarkdown		markdown: True | False | yes | NO
source	source	file	source	include
others	type = simple | multiline | grid | pipe			
		delimiter		
		quotechar		
		id (wrapped by div)		
Notes				width are auto-calculated when width is not specified

2 pantable2csv

This one is the inverse of `pantable`, a panflute filter to convert any native pandoc tables into the CSV table format used by pantable.

Effectively, `pantable` forms a "CSV Reader", and `pantable2csv` forms a "CSV Writer". It allows you to convert back and forth between these 2 formats.

For example, in the markdown source:

```
+--------+---------------------+--------------------------+
| First  | defaulted to be     | can be disabled          |
| row    | header row          |                          |
+========+=====================+==========================+
| 1      | cell can contain    | It can be aribrary block |
|        | **markdown**        | element:                 |
|        |                     |                          |
|        |                     | -   following standard   |
|        |                     |     markdown syntax      |
|        |                     | -   like this            |
+--------+---------------------+--------------------------+
| 2      | Any markdown        | $$E = mc^2$$             |
|        | syntax, e.g.        |                          |
+--------+---------------------+--------------------------+

: *Awesome* **Markdown** Table
```

running `pandoc -F pantable2csv -o output.md input.md`, it becomes

```
``` {.table}
---
alignment: DDD
caption: '*Awesome* **Markdown** Table'
header: true
markdown: true
table-width: 0.8055555555555556
width: [0.125, 0.3055555555555556, 0.375]
---
First row,defaulted to be header row,can be disabled
1,cell can contain **markdown**,"It can be aribrary block element:

-   following standard markdown syntax
-   like this
"
2,"Any markdown syntax, e.g.",$$E = mc^2$$
```
```