stitch Documentation

Release 0.3.3

Tom Augspurger

Sep 07, 2016

Contents

stitch is a library for making reproducible reports. It takes a markdown source file, executes the code chunks, captures the output, and stitches the output into the destination file.

Those familiar with knitr and RMarkdown will recognize it as a python clone of those great libraries. It's also heavily influenced by knitpy.

While `stitch` is written in python, it can be used for any of the dozens of Jupyter kernels.

CHAPTER 1

Examples

See the example site for a side-by-side comparison of the markdown source and rendered output. Links to a more complicated document rendered in various formats is also provided.

CHAPTER 2

Install

At the moment, the name stitch is taken on PyPI via an inactive project. You can install stitch from PyPI via

```
pip install knotr
```

I know, it's confusing. I've filed a claim for `stitch` on PyPI, but I think the people working that support queue are over-worked. Once that gets processed, I'll put it up on conda-forge as well. If you need a mnemonic, it's "I want knitr, but *not* the one written in *R*. Also I wanted to confuse R users. And knots are kind of like a buggy version of knits.

But to be clear the package name and command-line tool is `stitch`.

You'll also need to have a recent version of pandoc. Either use your system package manager, or use the `pypandoc` provided on conda-forge, which includes pandoc.

CHAPTER 3

Usage

You write a markdown file, and include code chunks that look like

```
```{kernel_name, [chunk_name], **kwds}
# your code here
```
```

The `kernel_name` is required (see `jupyter kernelspec list`). The `chunk_name` is optional; it controls things like the name assigned to plots if they're saved to disk.

The supported keyword arguments are

- eval: bool, whether to execute the code chunk
- echo: bool, whether to include the input code chunk in the output

More options will be added.

The command-line interface is essentailly the same as pandocs. For the most part you call

```
stitch input_file.md -o output_file.html
```

You can use `-t` for the output type, or infer it from the file extension of `-op`. All other options are passed through to pandoc.

stitch defines a few new options that control stitch-specific features

- `--no-standalone`
- `--no-self-contained`

CHAPTER 4

What's in a Name?

The name `stitch` has a couple meanings. Like R's `knit`, we are taking a source document, executing code chunks, and `knit`ing or `stitch`ing the output back into the document.

The second meaning is for `stitch` bringing together a bunch of great libraries, minimizing the work we have to do ourselves. `stitch` uses

- Pandoc markdown parsing and conversion to the destination output
- jupyter, specifically jupyter-client for managing kernels, passing code to kernels, and capturing the output
- pandocfilters for converting code-chunk output to pandoc's AST
- pypandoc for communicating with pandoc
- click for the command-line interface.

`stitch` itself is fairly minimal. The main tasks are

- processing code-chunk arguments
- passing the correct outputs from the jupyter kernel to pandocfilters
- assembling all the chunks of text, code, and output in a sensible way
- making things look nice

Contents:

4.1 Integration With Other Libraries

`stitch` doesn't make many assumptions, so it works pretty well with other libraries. For the most part, if it works in the notebook then it will work in `stitch`.

Note that source files all use the `.txt` extension so they open in your browser. Typically you'd use `.md` or `.markdown`.

- Bokeh: `source`, `HTML`
- Altair: `source`, `HTML`
- Holoviews: `source`, `HTML`

4.2 API

4.2.1 Chunk Options

Code chunks are blocks that look like

```
```{kernel_name, [chunk_name], **kwargs}
# code
```
```

The `kernel_name` is required, and `chunk_name` is optional. All parameters are separated by a comma.

kernel_name (*name: str*)
: Name of the kernel to use for executing the code chunk. Required. See `jupyter kernelspec list`.

chunk_name (*chunk_name: str*)
: Name for the chunk. Controls the filename for supporting files created by that chunk. Optional.

echo (*echo=True*)
: whether to include the input-code in the rendered output. Default True.

eval (*eval=True*)
: Whether to execute the code cell. Default True.

results (*s*)
: str; how to display the results

  •hide: hide the chunk output (but still execute the chunk)

warning (*True*)
: bool; whether to include warnings (stderr) in the ouput.

width (*w*)
: Width for output figure. See http://pandoc.org/MANUAL.html#images

> **Warning:** This will probably change to `fig.width` in a future release.

height (*w*)
: Height for output figure. See http://pandoc.org/MANUAL.html#images

> **Warning:** This will probably change to `fig.height` in a future release.

4.3 Whatsnew

4.3.1 Version 0.3.4

- API: Accept `fig.cap` as a chunk option to control the figure caption (GH38:)
- API: Exposed the `no-self-contained` command-line option to the stitching operation.
- API: Added a `warning` option for controling whether stderr is included in the output.

- API: Changed the `on_error` option to `error` for compatability with knitr and symmytry with the `warning` option.

4.3.2 Version 0.3.3

- Included `default.css` in the source and binary distributions (GH26).
- Fixed not handling output from IPython's various `display` methods (GH27).

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

C

E

H

K

R

W