

[R] RMarkdown vignettes v. Jupyter notebooks?

Ista Zahn [|@t@z@hn @ending from gm@||@com](mailto:istazahn@endingfromgmail.com)

Thu Oct 11 13:18:13 CEST 2018

- Previous message (by thread): [\[R\] RMarkdown vignettes v. Jupyter notebooks?](#)
- Next message (by thread): [\[R\] RMarkdown vignettes v. Jupyter notebooks?](#)
- Messages sorted by: [\[date\]](#) [\[thread\]](#) [\[subject\]](#) [\[author\]](#)

Hi Spencer,

On Thu, Oct 11, 2018 at 5:08 AM Spencer Graves

<spencer.graves@usingeffectivedefense.org> wrote:

>

> Hello:

>

>

> What are the differences between Jupyter notebooks and RMarkdown
> vignettes?

Here are some of the main differences I'm aware of:

Rmarkdown files include code and prose. The results produced by the code do not appear in the Rmarkdown file; instead, the file must be processed and typeset to produce a .pdf or .html (etc.) file that includes those results. Jupyter notebooks dispense with with processing step: output is displayed directly in the notebook. That is, a notebook includes code, prose, and results, while an Rmarkdown file includes only code and prose.

Rmarkdown allows control over the inclusion of code and results, and over how the results are displayed via header arguments. There is no such thing in Jupyter notebooks. Controlling the precise appearance of a document produced by Jupyter is much more difficult. Going farther, one can even say that Jupyter notebooks are designed primarily to be read as Jupyter notebooks; exporting to other formats is kind of an afterthought. In contrast, Rmarkdown is designed to produce the final readable result in another format (.html or .pdf typically).

Rmarkdown is based on markdown, a human readable markup language, Jupyter notebooks are based on JSON, a data interchange format common on the web. This means that Rmarkdown files can be easily edited using any text editor you like. The same is not true of Jupyter notebooks. While you can of course edit the JSON directly, the format is designed to be written and read by a computer; editing it yourself is not easy.

Rmarkdown is specific to R (I guess there is some basic support in knitr for other languages, but in my experience it never worked well) while Jupyter notebooks are language agnostic and "kernels" exist for a large number of programming languages. However, each Jupyter notebook can use only one kernel; you can't easily have R and Python code in the same notebook.

Jupyter notebooks typically run in your browser where the actual text editing features are somewhat limited. Rmarkdown is typically run in an editor such as Emacs or Rstudio where editing and project support is much better and greater customization may be possible. You can work indirectly with Jupyter notebooks in Emacs (<https://github.com/millejoh/emacs-ipython-notebook>) and perhaps other editors as well; this goes some way toward escaping the tyranny of the browser but is more fragile and difficult to get working compared to Rmarkdown.

Because Jupyter uses a web-based client-server model, it is easy to provide live interactive notebooks on your website (see e.g.,

<https://github.com/jupyterhub/binderhub>). As far as I know this is not currently possible with Rmarkdown.

>
>
> I'm trying to do real time monitoring of the broadcast quality of
> a radio station, and it seems to me that it may be easier to do that in
> Python than in R.[1] This led me to a recent post to
> "[python-list using python.org](https://python-list.using-python.org)" that mentioned "Jupyter, Mathematica, and the
> Future of the Research Paper"[2] by Paul Romer, who won the 2018 Nobel
> Memorial Prize in Economics only a few days ago. In brief, this article
> suggests that Jupyter notebooks may replace publication in refereed
> scientific journals as the primary vehicle for sharing scientific
> research, because they make it so easy for readers to follow both the
> scientific and computational logic and test their own modifications.

>
> A "Jupyter Notebook Tutorial: The Definitive Guide"[3] suggested
> I first install Anaconda Navigator. I got version 1.9.2 of that. It
> opens with options for eight different "applications" including
> JupyterLab 0.34.9, Jupyter Notebook 5.6.0, Spyder 3.3.1 (an IDE for
> Python), and RStudio 1.1.456.

>
>
> This leads to several questions:

>
> 1. In general, what experiences have people had with
> Jupyter Notebooks, Anaconda Navigator, and RMarkdown vignettes in
> RStudio, and the similarities and differences? Do you know any
> references that discuss this?

I've used both extensively, and noted the differences I've discovered above.

>
>
> 2. More specifically, does it make sense to try to use
> RStudio from within Anaconda Navigator, or is one better off using
> RStudio as a separate, stand alone application -- or should one even
> abandon RStudio and run R instead from within a Jupyter Notebook? [I'm
> new to this topic, so it's possible that this question doesn't even make
> sense.]

The only advantage I can think of to using Rstudio via Anaconda is that you could use conda environments to maintain different versions of R and/or R packages for different projects.

You'll have to weigh the pros and cons to decide whether to switch from Rstudio to Jupyter notebooks. Depending on what you want to do there are both advantages and disadvantages, as discussed above.

Finally, I have to give a plug for a couple of related tools that I find very useful.

Emacs org-mode <https://orgmode.org/> gives you the best of both worlds: notebooks unconstrained by the browser that can include code in multiple languages, header arguments, excellent export support, etc. It is superior to both Jupyter and Rmarkdown, except that support only exists in Emacs.

Jupytertext (<https://github.com/mwouts/jupytertext>) is another way to have it all, by allowing you to edit in markdown or Rmarkdown, and auto-generating a notebook and possibly other formats for you. I've only recently started experimenting with it, but so far I like it a lot.

Best,
Ista
>
>

> Thanks,
> Spencer Graves
>
>
> [1] If you have ideas for how best to do real time monitoring of
> broadcast quality of a radio station, I'd love to hear them. I need
> software that will do that, preferably something that's free, open
> source. The commercial software I've seen for this is not adequate for
> my purposes, so I'm trying to write my own. I have a sample script in
> Python that will read a live stream from a radio tuner and output a
> *.wav of whatever length I want, and I wrote Python eight years ago for
> a similar real time application. I'd prefer to use R, but I don't know
> how to get started.
>
>
> [2] 2018-04-13:
> "<https://paulromer.net/jupyter-mathematica-and-the-future-of-the-research-paper>".
> This further cites a similar article in The Atlantic from 2018-04-05:
> "www.theatlantic.com/science/archive/2018/04/the-scientific-paper-is-obsolete/556676".
>
>
> [R-help using r-project.org](#) mailing list -- To UNSUBSCRIBE and more, see
> <https://stat.ethz.ch/mailman/listinfo/r-help>
> PLEASE do read the posting guide <http://www.R-project.org/posting-guide.html>
> and provide commented, minimal, self-contained, reproducible code.

- Previous message (by thread): [\[R\] RMarkdown vignettes v. Jupyter notebooks?](#)
 - Next message (by thread): [\[R\] RMarkdown vignettes v. Jupyter notebooks?](#)
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)
-

[More information about the R-help mailing list](#)