

nbinteract: Generate Interactive Web Pages From Jupyter Notebooks

Samuel Lau

April 16, 2018

Abstract:

Introduction

Jupyter notebooks provide a popular document format for authoring, executing, and publishing code alongside analysis [12]. Although Jupyter notebooks were originally designed for use in scientific workflows for data preparation and analysis, they are becoming an increasingly common choice for university courses—a survey in 2016 reported that over one hundred courses across multiple countries use Jupyter in their course content [7].

An increasing number of universities now offer data science courses, many of which use Jupyter because of its broad adoption for data analysis workflows in both academia and industry. These courses often use Jupyter notebooks as the preferred medium for homeworks, labs, projects, and lectures. UC Berkeley’s flagship data science courses, for example, use Jupyter for all of these course components and have even written their course textbooks in Jupyter notebooks.

As a web technology, Jupyter notebooks also provide a platform for interaction authoring. For example, the popular `ipywidgets` Python library allows users to create web-based user interfaces to interact with arbitrary Python functions. Users can create these interfaces using Python directly in the notebook environment instead of having to use HTML and Javascript, significantly lowering the overhead usually needed to create these interfaces [9]. This ease-of-use encourages instructors and researchers to create interactive explanations of their work.

Unfortunately, it is difficult to share these interactive notebooks with a broad audience. Sharing the notebook file itself retains full interactivity but requires viewers to have Jupyter, Python, and all other packages used in the notebook installed on their own machines. The freely available Binder service circumvents this by hosting notebook servers that come pre-packaged with necessary software. However, both of these options still require viewers to have prior familiarity with the Jupyter environment, making them less suitable for use with non-technical viewers. Authors can convert a Jupyter notebook to a static HTML document and host the document as a publicly-

accessible web page. However, this method does not preserve the interactive elements of the notebook; the resulting web page only contains text and images.

`nbinteract` is a Python package that allows authors to convert Jupyter notebooks into interactive HTML pages. The interactive elements can use arbitrary Python code to generate output, including Python libraries that use C extensions (e.g. `numpy` and `pandas`) and libraries that create images (e.g. `matplotlib`). The resulting web pages can be used by anyone with a modern web browser even if the viewer does not have Python or Jupyter installed on their computer. The `nbinteract` package also includes specialized methods for interactive plots designed for fast interaction prototyping in the notebook and smooth interaction on static HTML web pages. We discuss the design of the package, its features and limitations, and its implications for interaction authoring and sharing.

Related Work

Jupyter Technologies

The Jupyter notebook platform allows authoring and editing code, images, and written explanations together in a single document composed of multiple cells. The platform is composed of two main components. It includes a frontend—a web-based authoring environment that users open in their web browsers. The frontend connects to a Jupyter kernel, a process on the users’ computers that runs code and returns the output to the frontend to display [12].

The `ipywidgets` library makes use of Jupyter’s web-based frontend to create interactive elements directly in the notebook. The library includes Python functions that produce HTML and Javascript when called to create widgets. When a user interacts with a widget—selecting an option from a dropdown menu, for example—the `ipywidgets` library executes user-defined Python functions on the Jupyter kernel and renders the result in the cell [9]. A number of other specialized libraries are built on top of `ipywidgets`, such as the interactive plotting library `bqplot` [4] and the molecular visualization library `nglview` [1].

Jupyter notebooks use the `nbconvert` tool to convert between notebook formats. `nbconvert` also allows notebooks to be converted to static HTML pages [8]. However, these pages do not retain widget functionality because they do not have access to a Jupyter kernel by default.

The Binder project hosts ephemeral Jupyter notebook servers as a free service for the general public. It takes a repository of Jupyter

notebooks, starts a Jupyter frontend and Jupyter kernel, and gives users the ability to run the notebook over the internet instead of having on their local machines [2].

Interaction Authoring in Javascript

Javascript is the most commonly used language to design interactions that run in a web browser. Because most modern web browsers run Javascript natively, viewers do not have to install additional software in order to make use of these interactive elements, a key advantage of the language. A number of authors use Javascript to create interactive articles [6, 10] and textbooks [11].

There are a number of Javascript libraries that provide higher level abstractions for interaction creation such as D3 and Tangle [3, 5]. Fundamentally, Javascript libraries require fluency with aspects of web programming such as Javascript syntax and the document-object model. This additional requirement makes Javascript more difficult to use for many data scientists; most data science analysis uses Python and R.

References

- [1] Arose/Nglview: IPython Widget to Interactively View Molecular Structures and Trajectories. <https://github.com/arose/nglview>.
- [2] Binder (Beta). <https://mybinder.org/>.
- [3] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. “Data-Driven Documents”. In: *IEEE transactions on visualization and computer graphics* 17.12 (2011), pp. 2301–2309.
- [4] Bqplot: Plotting Library for IPython/Jupyter Notebooks. Apr. 2018.
- [5] bret. Tangle: A JavaScript Library for Reactive Documents. Apr. 2018.
- [6] Explorable Explanations. <http://explorables.es/>.
- [7] Jessica B. Hamrick and Jupyter Development Team. 2016 *Jupyter Education Survey*. May 2016. DOI: 10.5281/zenodo.51701.
- [8] Jupyter/Nbconvert: Jupyter Notebook Conversion. <https://github.com/jupyter/nbconvert>.
- [9] Jupyter-Widgets/Ipywidgets: Interactive Widgets for the Jupyter Notebook. <https://github.com/jupyter-widgets/ipywidgets>.
- [10] Daniel Kunin. *Seeing Theory*. en. <http://seeingtheory.io>.
- [11] Philip B. Stark. “SticiGui: Statistics Tools for Internet and Classroom Instruction with a Graphic User Interface”. In: *Available at Web site: http://stat-www.berkeley.edu/users/stark/SticiGui/index.htm (8 August 2004)* (2004).
- [12] Kluyver Thomas et al. “Jupyter Notebooks—a Publishing Format for Reproducible Computational Workflows”. In: *Stand Alone* (2016), pp. 87–90. ISSN: 0000-0000. DOI: 10.3233/978-1-61499-649-1-87.