

PsiPyPublish: An IPyPublish Template for Psychological Research

Stefan Uddenberg

Princeton University

Running head: PsiPyPublish

Address for correspondence:
Stefan Uddenberg
Peretsman Scully Hall 322
Princeton University
Princeton, NJ 08540
stefanu@princeton.edu

Word count: XXX (Main text + abstract)

Version: Template file – not for submission

February 1, 2019

Contents

1	Introduction	3
1.1	Configuration	3
2	Notes	4
2.1	Production	4
2.2	Markdown	4
2.3	Embed HTML, including video	4
2.4	Citations and References	5
2.5	Figures	5
2.6	Templating — Pass Variables into Markdown	6
2.7	Latex	6
2.8	Terminal commands	6
3	References	8

1 Introduction

This is a template for an APA-style [iPyPublish](#) manuscript. Feel free to check out the documentation and examples at that link; it's all very good. There you can find information on how to embed figures, code, tables, and more. References are managed using [Zotero](#) in concert with [Better BibTex](#). For now, you're going to want to edit the notebook's metadata in order to change what appears on the title page. In addition, the metadata includes `jupyter` configuration, so that you can automatically generate markdown and `py:percent` versions of this notebook automatically on saving -- assuming you have `jupyter` installed and correctly configured, that is!

1.1 Configuration

My working configuration files for Jupyter (with Jupyter) and iPyPublish can be found in this repository. Naturally, you will need to replace your computer's original versions of these files with the new ones included here. For example, if using Anaconda, your iPyPublish installation can be found at `your_environment_name/Lib/site-packages/ippublish`.

2 Notes

2.1 Production

Produce a notebook in the terminal with the command `nbpublish -pdf -pbug file_name.ipynb`¹. Outputs to converted folder at the `.ipynb` file's location.

2.2 Markdown

- Headings and sub-headings (and so on) are made by prefacing text with `#`. The more `#`s, the greater the level of heading.
- Unordered lists are made by prefacing text with a `"-"`.
 1. Numbered lists start with a number and dot.
 2. Create sublists via tabbed indentation.
- Footnote links are made with `[^X]` (where `X` is some number). Footnote content is placed below with `[^X]: Content goes here`. Here's an example.²
 - Correct formatting only appears after running `nbpublish`.
- [Links](#) can be generated with the following syntax: `[link](http://www.website.com)`
- Code can be placed between backticks (the character to the left of the `1` key at the top of your keyboard).
 - Place it between 3 backticks (with an optional language name) and you get (syntax-highlighted) block code.³ `python print(foo)`
- *Italic*, **bold**, and ***bolded italic*** text can be created by sandwiching text between 1, 2, or 3 `*`s or `_`s respectively.
- Blockquotes are made by prefacing text with `>` .

Get inline todos with Latex's "todo" command.

2.3 Embed HTML, including video

HTML embedding is accomplished via the `%%HTML` cell magic. Naturally, this won't appear in a PDF export.

¹ Technically `-pbug` is optional so you can see verbose output, but `nbpublish` seems to work more reliably with this option enabled.

² Footnote content goes here!

³ Note, however, that one should not use this for displaying large chunks of code in an `nbpublish` PDF. Instead, see code cell [code 2.1](#) below for an example of how to place code in the PDF

2.4 Citations and References

- First, specify the bibliography entry in the notebook metadata to the correct bibliography file (Edit --> Edit Notebook Metadata). *Leave out the .bib extension from this file name!* It should look like `path/to/bibFileName`.
 - If nbpublish is having problems finding the .bib file, I have had success by placing a copy in the `converted/notebook_name_files/` directory, as well as placing the file in the same folder as the actual notebook. This makes set up for the notebook's bibliography metadata especially easy.
- Citations can be input with citation keys and standard latex commands (e.g., `\cite{citationKey}`).
- I've had success with citation keys generated via Zotero Better BibTex, like so (Uddenberg & Scholl, 2018). Note that you won't see the final formatted output until you run nbpublish.
- See a [cheat sheet of valid cite commands here](#).

2.5 Figures

- Figures can be displayed with commands like `display(SVG("filename.svg"))` or `Image('filename.jpg', height=400)`.
- Edit the cell's metadata to change the figure caption, placement, size, et al. (View --> Cell Toolbar --> Edit Metadata --> Click on "Edit Metadata" above cell.)
- They can be linked in markdown via `\cref{fig:figNameFromMetadata}`

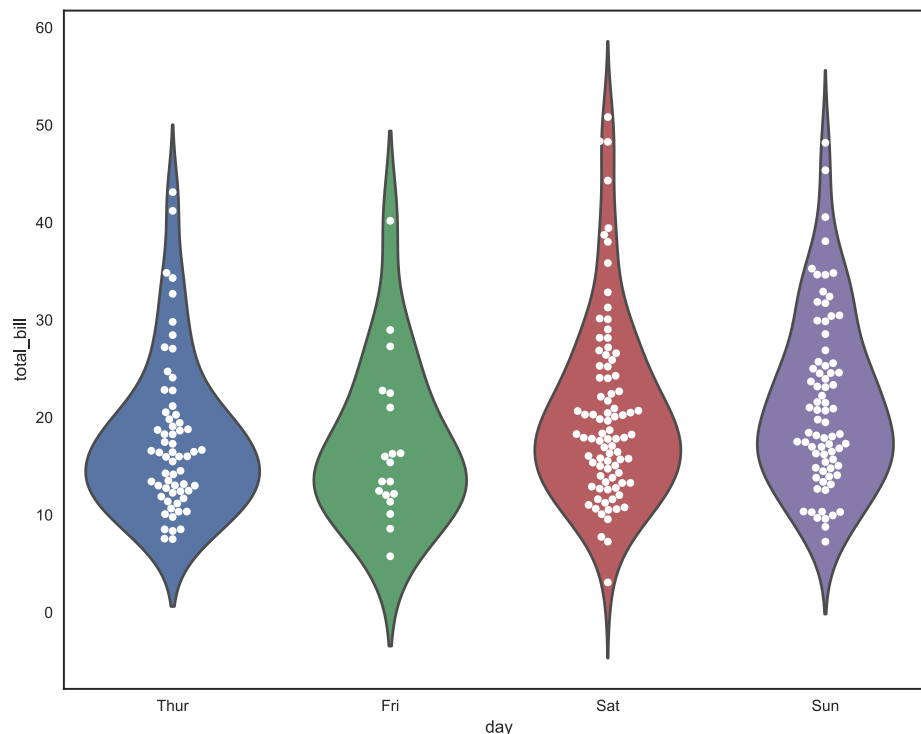


Figure 2.1: An example beeswarm plot of Seaborn’s built-in ‘tips’ dataset.

2.6 Templating — Pass Variables into Markdown

- Pipe valid Python code into markdown directly by sandwiching it between two curly braces: E.g., $2 + 2 = 4$
- Note that the notebook needs to be Trusted; look to the top right to see if it is and simply click on Not Trusted to change that.

2.7 Latex

- Execute arbitrary latex by sandwiching it between dollar signs: $a = b + c$
- Alternatively, use `Latex()` command from `ipypublish` within a code cell.
- Latex’s `hphantom` command is useful when you just want a little more horizontal space between items.

2.8 Terminal commands

- Execute terminal commands in Jupyter by prefacing code with `!`.
- For example, you can export this notebook with the following code cell:

Code 2.1: How to publish this notebook, from within the notebook itself!

```
1 !nbpublish -pdf -pbug template.ipynb
```

3 References

Uddenberg, S., & Scholl, B. J. (2018). Teleface: Serial reproduction of faces reveals a whiteward bias in race memory. *Journal of Experimental Psychology: General*, 147(10), 1466-1487.