

First Notebook War

PyData DC 2018

Martin Skarzynski

2018-11-17

About me

- Cancer Prevention Fellow
 - at National Cancer Institute
- Co-Chair, Bioinformatics & Data Science
 - at Foundation for Advanced Education in the Sciences
- Website: <https://marskar.github.io>
- Twitter: [@marskar](#)



Title inspired by a tweet by Philip Guo



Previous conflicts

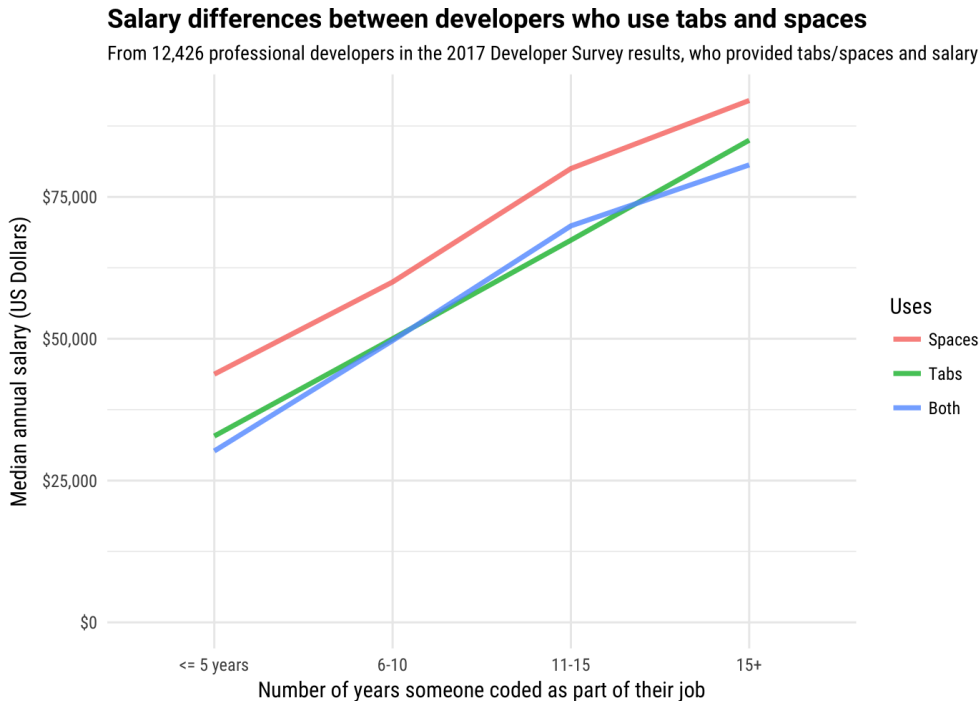
- Spaces versus Tabs
- Emacs versus Vim



- Source: <https://goo.gl/images/U2KpcG>

Spaces versus Tabs

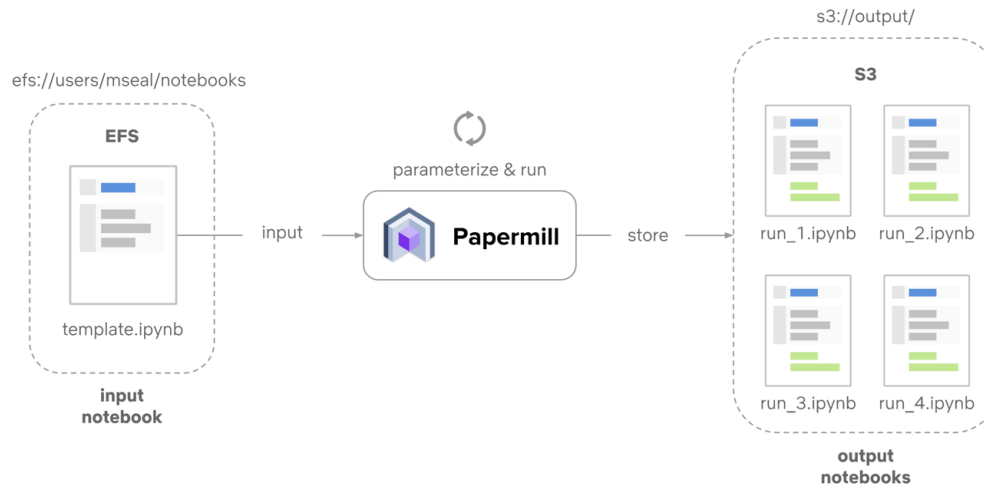
- Code editor setup: Tab = 4 spaces
- **GNU Make** requires tabs!
- Use spaces, get paid more!
 - according to **blog post** by **David Robinson (@drob)**'s



Notebooks

Jupyter notebooks

- are data science tools
- built on IPython by **Fernando Perez (@fperez)**
- combine **Markdown** text, code, and output
- help data scientists communicate goals, methods, and results
- used in **academia**, **Amazon**, **Netflix**, and **PayPal**



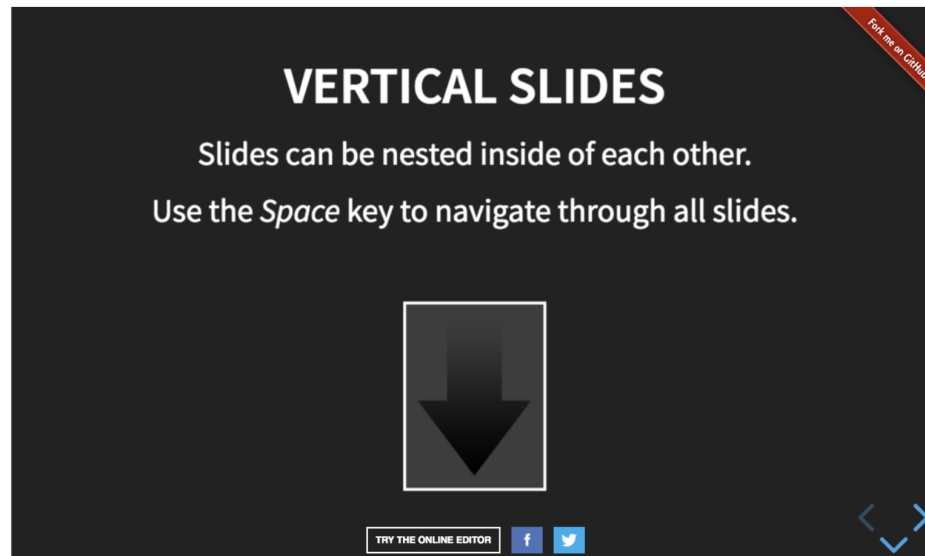
Joel doesn't like notebooks

- "I Don't Like Notebooks" by Joel Grus (@joelgrus) at Jupytercon 2018
- [Slides](#)
- [Video](#)
- Modularity and Reusability

Joel also doesn't like vertical slides

PLEASE COME TO MY NEXT TALK:

"I DON'T LIKE THAT SLIDESHOW PROGRAM WHERE SOMETIMES THE NEXT SLIDE IS RIGHT AND SOMETIMES THE NEXT SLIDE IS DOWN"



@joelgrus #jupytercon

Tim Hopper (@tdhopper) likes notebooks



Supersnazz Carlson @scottythered · Nov 13

Replying to @tdhopper

Tim, you know what I like best about this, aside from the sentiment?
It's not 150 slides long.



Joel Grus @joelgrus · Nov 13

I also find its brevity its most compelling aspect



Tim Hopper @tdhopper · Nov 13



DataFramed

- Podcast by Hugo Bowne-Anderson (@hugobowne)
- Episode 44 with Brian Granger (@ellisonbg)
- JupyterLab



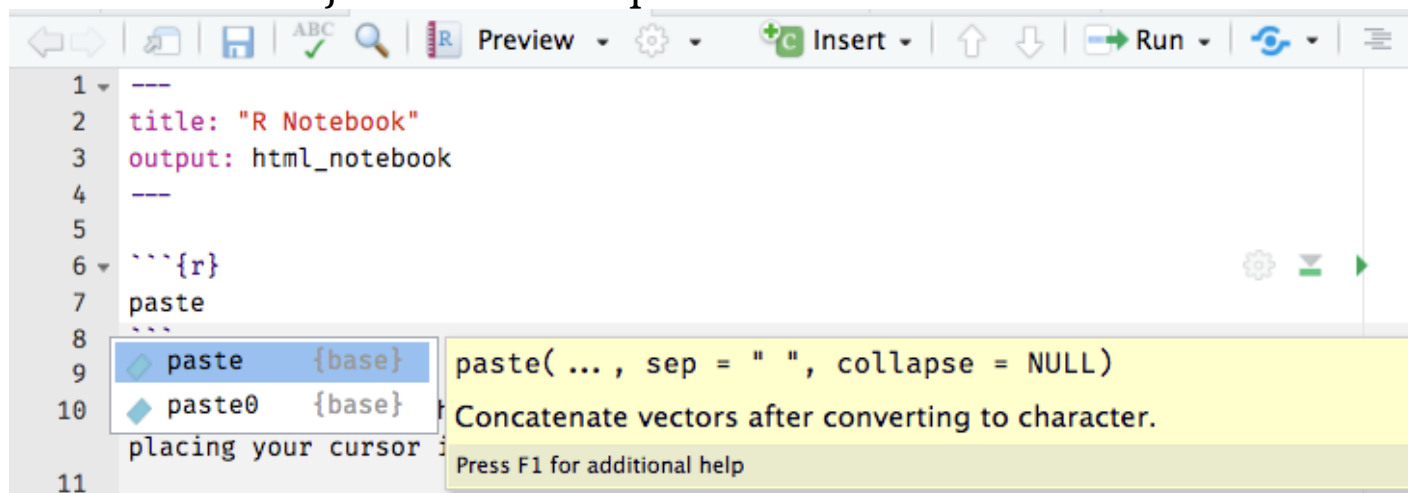
« We are entering an era where large, complex organizations need to scale interactive computing with data to their entire organization in a manner that is collaborative, secure, and human centered. »

BRIAN GRANGER



Yihui Xie (@xieyihui)'s Blog post

- I used **R markdown** make these slides!
- No problems with version control
- **R notebooks** are just another output format!



The screenshot shows the R Notebook editor interface. The top toolbar includes icons for navigation, saving, previewing, and running. The code editor displays a YAML header for an R Notebook and a code chunk. A tooltip is visible for the `paste` function, showing its syntax and description.

```
1 ---  
2 title: "R Notebook"  
3 output: html_notebook  
4 ---  
5  
6 ```{r}  
7 paste  
8 ```  
9  
10  
11
```

paste {base}
paste0 {base}

paste(... , sep = " ", collapse = NULL)
Concatenate vectors after converting to character.
Press F1 for additional help

Why use notebooks

- Literate programming
- Rendered by GitHub and nbviewer
- Google colab
- Kaggle kernels
- Binder

Notebook tools

1. version control tool for notebooks - **nbdime**
2. work with Jupyter notebooks and scripts in parallel using **JupyterText**
3. configure notebooks to run on markdown (md) files with **notedown**
4. create and run Jupyter notebooks from scripts and md files with **nbless**

A quick insight at world population

Collecting population data

In the below we retrieve population data from the [World Bank](#) using the [wbdata](#) python package

```
In [1]: import pandas as pd
import wbdata as wb

pd.options.display_max_rows = 6
pd.options.display_max_columns = 20
```

Corresponding indicator is found using search method - or, directly, the World Bank site.

```
In [2]: wb.search_indicators('Population, total') # SP.POP.TOTL
# wb.search('in indicators', 'area')
# => https://data.worldbank.org/indicator is easier to use
```

SP.POP.TOTL Population, total

Now we download the population data

```
In [3]: indicators = {'SP.POP.TOTL': 'Population, total',
                    'AG.SRF.TOTL.K2': 'Surface area (sq. km)',
                    'AG.LND.TOTL.K2': 'Land area (sq. km)',
                    'AG.LND.ARBL.ZS': 'Arable land (% of land area)'}
data = wb.get_dataframe(indicators, convert_date=True).sort_index()

Out[3]:
```

country	date	Population, total	Surface area (sq. km)	Land area (sq. km)	Arable land (% of land area)
Afghanistan	1980-01-01	8996311.0	NaN	NaN	NaN
Algeria	1980-01-01	1918111.0	919585.0	652861.0	73.16
Algeria	1962-01-01	9345888.0	919585.0	652861.0	71.794259

Traditional Jupyter notebook

Write modules!

- Imports
 1. Standard Library
 2. Third Party
 3. User Defined
- Definitions
 - Classes
 - Functions (for more check out Steven Lott's PyData DC tutorial)
- Type Hints
- Docstrings (with examples!)
- `if __name__ == '__main__':`
- Function call(s), e.g. `doctest.testmod(verbose=True)`
- `doctest`: docstring examples -> test suite (with `unittest API`)
- run test suite with `unittest` or `pytest`
- use `cookiecutter` for project structure
- `deploy projects/packages to PyPI`

Thanks for listening!