Computational Notebooks

Huq Imdadul, Memmel Marius
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Definition</td>
<td></td>
</tr>
<tr>
<td>2. What are computational notebooks?</td>
<td></td>
</tr>
<tr>
<td>3. Why use computational notebooks?</td>
<td></td>
</tr>
<tr>
<td>4. Use cases</td>
<td></td>
</tr>
<tr>
<td>5. What’s wrong about computational notebooks?</td>
<td></td>
</tr>
<tr>
<td>6. Conclusion / discussion</td>
<td></td>
</tr>
</tbody>
</table>
Definition

Literate Programming

‘I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature.’


‘[Literate programming] pairs the functionality of word processing software with both the shell and kernel of [a] notebook’s programming language.’

- Wikipedia, Notebook Interface [3]
Definition

Computational Notebook

‘A notebook interface (also called a computational notebook) is a virtual notebook environment used for literate programming.’

- Wikipedia, Notebook Interface [3]

Mixed Notebooks

‘[Mixed notebooks are a] new generation of notebooks that is based on cells, each of which contains rich text or code that can be executed to compute results or generate visualizations.

- Exploration and Explanation in Computational Notebooks [12]
Some Examples

- databricks
- Amazon SageMaker
- jupyter
- WolframAlpha
- Spark
- colab
- Kaggle
Technology at the example of Jupyter Notebooks

- **Frontend**: code editor
- **Kernels**: computational engines
- Communication via API

--> Separation of content and execution

--> Multi-language support by swapping kernels
Template

Narrative Text

Notebook title and introduction

Description of model parameters

Description of need to profile data

Code and Visualizations

Importing external packages

Implementation of parameters

Profile plotting code

Inline plot
Data science is an iterative exploratory process of extracting insights from data.

Assumptions / situations

- Small changes can lead to different results --> documentation essential
- Iterative and exploratory approach --> difficult documentation
- ‘Dead ends’
- Process creates many figures, files and scripts with similar names
Computational notebooks to the rescue!

- Combination of code, text and visualizations in a single document
- Easy to share
- Easy to iterate fast and debug code

→ Enables quick prototyping and EDA
And they can do even more …

- Cloud offers
- Platform independence
- Computational narrative
- Single document
- Reproducibility
- ...
Use Cases

Education:
- Coding tutorials
- Data analysis
- Visualization (techniques)

Commercial:
- distill.pub
- Netflix
distill.pub: modern medium for presenting research
Netflix: reimagining notebooks

- Unified tool for most common data jobs
  - Run code, explore data, present results

- Use cases
  - Data access
  - Notebook templates (parameterization)
  - Scheduling notebooks
Netflix: scheduling notebooks
What's wrong about Computational Notebook?

- **Fundamental idea of notebook**
  - Quick input for a single step, get fast feedback, share
  - … & iterate

- **Negative effects**
  - Leads to bad practices -> Encourages polluting global space, discourage code reusability….
  - Like a junk food, if eaten too much it makes you obese & harder to maintain
  - Number of pain points
What's wrong about Computational Notebook?

9 Pain points[7].

- **Setup**
  - Repeating tasks like external loading & cleaning heavy data.
  - Also sometime leads to crash.

- **Explore and analyze**
  - Modeling & visualization at the same time is frustrating.
.... 9 Pain points.

- Manage code
  - Not an IDE, missing autocomplete, documentation, package dependencies

- Reliability
  - Occasional crash -> No feedback -> Inconsistent state = Makes it unreliable.
  - Resulting restarting notebook & iterate the process again. Especially with Big Data.

- Archival
  - No out-of-the-box version controlling system.
9 Pain points.

- **Security**
  - No masking to sensitive data while sharing notebook to execute.
  - No read-only or run-only feature.
  - External tools required for enforce access.

- **Share & Collaborate**
  - Share data, documentation for setup is needed.
  - Sharing with non-technical person is not supported.
… 9 Pain points.

- **Reproduce & Reuse**
  - Because of dependency & environment setting ability to reproduce & reuse is difficult.

- **Notebooks as product.** Deploying to production requires significant cleanup & packaging of libraries - Outside of core skills of data scientist.
Good Software engineering?

Rigorous software engineering isn't that important, I'm just experimenting!

You mean you're just doing science?

I just want to see if my model works before I put it into production.

Don't you need to write correct code to make sure it works?

Not in the best Balance

src-https://docs.google.com/presentation/d/1n2RIMdmv1p25Xy5thJUhkKGvjtV-dkAlsUXP-AL4ffI
Tools for reducing pain

- **nbdime.** Jupyter Notebook Diff and Merge tools

```bash
$ diff a.ipynb b.ipynb
76,77d75
< "plt.rc(axes', grid=False)"
< "plt.rc(axes', facecolor='white')"
90c88
< "image/png": "iVBORw0KGoAAAANSUhEUgAAABLkAAAMQCAAYA/AAAAALw5FLz\nAAAMJQAAFiUB5VIk8AAATIBJREFUEj3zsXeY2F57b12h6
```

![Image of code output](image-url)
Tools for reducing pain

- nbgather

Prologue: Making a Mess
More tools

- **Papermill**. A tool for parameterizing and executing Jupyter Notebooks. It can store output notebooks cloud storages.

- **interact** is an open-source, desktop-based, interactive computing application

- **NbExtensions** provides a collection of unofficial extensions for use with Jupyter Notebook. Some of the extensions provided, allow convert python 2 to python 3 code, push to github gist, automatic code formatting etc.
Statistics from Github on Notebook usage

Conclusion

- Great for data scientists to quickly data analyzation and fast iterations
- Questionable software engineering technique when it comes to maintainability, reliability & shipping to production
- Number of external tools available who try to solve the shortcomings
- If discipline is maintained, they are an effective toolbox
THANK YOU EVERYONE
Discussion

What do you think, is notebook suitable for production?
Discussion

Pro or con computational notebook?
Reference

[1] https://netflixtechblog.com/notebook-innovation-591ee3221233
[8] https://docs.google.com/presentation/d/1n2RIImdmv1p25Xy5thJUhkKGvjtV-dkAIsUXP-AL4ff
[10] https://github.com/microsoft/gather
License

Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

https://creativecommons.org/licenses/by-sa/4.0/