Managing Messes in Computational Notebooks

Andrew Head · Fred Hohman · Titus Barik · Steven M. Drucker · and Robert DeLine

UC Berkeley · Georgia Tech · Microsoft Research
Computational Notebooks: Code, Text, and Output

Rich descriptions

Code

Output
Notebook Programming Interfaces Abound

I saw it reliably provides top-rate insights, humor, or both. I was thrilled when I got to introduce Randall Monroe for a talk in 2007. But in skod #1313.

```python
from __future__ import division, print_function
import re
import itertools

def words(text): return set(text.split())


We can see that there are multiple names that are both winners and losers:

```
In [2]: winners & losers
Out[2]:
{'adams', 'bush', 'carter', 'cleveland', 'clinton', 'harrison', ...
```
1. Incremental execution
1. Incremental execution
2. In-situ output
Notebook Model of Exploratory Programming

1. Incremental execution
2. In-situ output
3. Incremental changes
### Notebook Model of Exploratory Programming

1. **Incremental execution**
2. **In-situ output**
3. **Incremental changes**
4. **Control over layout**

```python
In [1]: import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn import datasets

In [2]: data = datasets.iris().data[:,:4]
   :   petal_length, petal_width = data[:,0], data[:,1]

In [12]: petal_length, petal_width = data[:,1], data[:,0]

In [3]: print("Average petal length: %.3f" % (sum(petal_length) / len(petal_length),))
   :   Average petal length: 3.758

In [10]: clusters = KMeans(n_clusters=2).fit(data).labels_

In [11]: plt.scatter(petal_length, petal_width, c=clusters)

Out[11]: <matplotlib.collections.PathCollection at 0x10b32ea20>
```
Notebook Model of Exploratory Programming

1. Incremental execution
2. In-situ output
3. Incremental changes
4. Control over layout
1. Incremental execution
2. In-situ output
3. Incremental changes
4. Control over layout
Notebook Model of Exploratory Programming

1. Incremental execution
2. In-situ output
3. Incremental changes
4. Control over layout

How did I produce this?

1 WEEK LATER

1. How did I produce this result?
Notebook Model of Exploratory Programming

1. Incremental execution
2. In-situ output
3. Incremental changes
4. Control over layout

1 WEEK LATER

How did I produce this result?

```python
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn import datasets

data = datasets.load_iris().data[:,2:4]
petal_length, petal_width = data[:,0], data[:,1]
print("Average petal length: %.3f" % (sum(petal_length) / len(petal_length)))
Average petal length: 3.758
```
1. Incremental execution
2. In-situ output
3. Incremental changes
4. Control over layout

1 WEEK LATER
1. How did I produce this result?
2. Didn't I have a better version of this?
1. Incremental execution
2. In-situ output
3. Incremental changes
4. Control over layout

1 WEEK LATER
1. How did I produce this result?
2. Didn't I have a better version of this?
3. What can I get rid of?
Messes in Computational Notebooks

Disappearance
Deleted / overwritten code

Disorder
Out-of-order execution
1/2 of notebooks on GitHub [Rule et al. 2018]

Dispersion
Too many cells

Notebooks contain ugly code and dirty tricks [Rule et al. 2018]

31 / 41 surveyed participants had trouble finding prior analyses [Kery et al. 2018]
Managing Messes in Computational Notebooks

How can tools help analysts find, recover, and compare code in messy notebooks?

| [1] | How messes happen |
| [∗] | **CODE GATHERING TOOLS** |
| [ ] | Tools in context |
| [ ] | Implementation |
| [ ] | Qualitative usability study |
CODE GATHERING TOOLS Demo

1 WEEK Passes
Task 1: Recovering Code

How did I produce this?
Task 1: Recovering Code

How did I produce this?
Task 1: Recovering Code

How did I produce this?
**CODE GATHERING TOOLS Demo**

Task 1: Recovering Code

How did I produce this?

Request cell subset that produced the result.
Code Gathering Tools Demo

Task 1: Recovering Code

How did I produce this?

Request cell subset that produced the result.
**Code Gathering Tools Demo**

Task 1: Recovering Code

How did I produce this?

The gathered code is...
- reduced
- ordered
- complete

Request cell subset that produced the result.
**CODE GATHERING TOOLS Demo**

**Task 1: Recovering Code**
Request cell subset that produced the result.

**Task 2: Comparing Versions**

*Didn't I have a better version of this?*
**Code Gathering Tools Demo**

**Task 1: Recovering Code**
Request cell subset that produced the result.

**Task 2: Comparing Versions**

*Didn't I have a better version of this?*

Open a version browser for a result.
Task 1: Recovering Code
Request cell subset that produced the result.

Task 2: Comparing Versions

Didn't I have a better version of this?

Open a version browser for a result.
CODE GATHERING TOOLS Demo

Task 1: Recovering Code
Request cell subset that produced the result.

Task 2: Comparing Versions

Didn't I have a better version of this?

Open a version browser for a result.
Task 1: Recovering Code

Request cell subset that produced the result.

Task 2: Comparing Versions

Didn't I have a better version of this?

Open a version browser for a result.
**Code Gathering Tools Demo**

**Task 1: Recovering Code**

Request cell subset that produced the result.

**Task 2: Comparing Versions**

Didn't I have a better version of this?

Open a version browser for a result.
### Code Gathering Tools Demo

#### Task 1: Recovering Code
Request cell subset that produced the result.

#### Task 2: Comparing Versions
Open a version browser for a result.

#### Task 3: Cleaning Notebook
What code can I get rid of?
Task 1: Recovering Code
Request cell subset that produced the result.

Task 2: Comparing Versions
Open a version browser for a result.

Task 3: Cleaning Notebook
What code can I get rid of?

... Request cell subset that produced the result.
How can tools help analysts manage messes in their notebooks?

**Task 1: Recovering Code**

Request cell subset that produced the result.

**Task 2: Comparing Versions**

Open a version browser for a result.

**Task 3: Cleaning Notebook**

... Request cell subset that produced the result.
Post-Hoc Mess Management

Helping analysts clean and navigate their code whether or not they adopted a strategy to version or organize their code.
Managing Messes in Computational Notebooks

How can tools help analysts find, recover, and compare code in messy notebooks?

[1] How messes happen


[*] Implementation

[*] Qualitative usability study
Implementation: Slicing Notebooks

1 Notebook
some cells missing,
some cells out-of-order

? cleaned, ordered notebooks

versioned results
Implementation: Slicing Notebooks

① Notebook
some cells missing, some cells out-of-order

② Execution Log
all cells present, in-order
Implementation: Slicing Notebooks

1. Notebook
   - some cells missing,
   - some cells out-of-order

2. Execution Log
   - all cells present, in-order

 execution time

[10] [11] [1] [12] [3]
Implementation: Slicing Notebooks

1. Notebook
   - some cells missing, some cells out-of-order

2. Execution Log
   - all cells present, in-order

3. Program Slices [Weiser '81]
Implementation: Slicing Notebooks

1. Notebook
   - some cells missing,
   - some cells out-of-order

2. Execution Log
   - all cells present, in-order

3. Program Slices [Weiser '81]
   - which can be used to make...
     - cleaned, ordered notebooks (preserve cell boundaries and outputs)
     - versioned results (slice all cell versions)
Cleaning and Exploring Messy Notebooks
A Sample of Recent Research

Interactions for Untangling Messy History in a Computational Notebook
Kery et al., VL/HCC '18

Towards Effective Foraging by Data Scientists to Find Past Analysis Choices
Kery et al., CHI '19

Aiding Collaborative Reuse of Computational Notebooks with Annotated Cell Folding
Rule et al., CSCW '18

Design and Use of Computational Notebooks
Rule, Ph.D. Thesis, '18
Evaluating Code Gathering Tools

Q1. What is the meaning of "cleaning"?

Q2. How do analysts use code gathering tools during exploratory data analysis?
A Qualitative Study of Gathering

Participants: $N = 12$ professional data analysts

Cleaning Task $\times 2$: Clean a computational notebook, with and without code gathering tools.

Exploration: Rank movies in from a movies dataset. Use code gathering tools as you wish.
Q1. The Meaning of "Cleaning"

Picking a subset of cells [P1-P12]...
and removing the rest [P8, P10-12].

"I picked a plot that looked interesting and, if you think of a dependency tree of cells, walked backwards and removed everything that wasn’t necessary."

... And many additional stages:

polishing visualizations [P1, P6]  restructuring code [P3, P4, P6, P12]
integrating with version control  [P7]
Q2. How do analysts use code gathering tools during exploratory data analysis?

Participants described gathering to a notebook as "beautiful" and "amazing": it "hits the nail on the head."
Some Observed Uses of Gathering Tools

"Finishing moves"

Gathering for multiple audiences

Lightweight branching

Creating personal references
Takeaways from Study

Q1. Gathering covers an important *yet incomplete* set of notebook cleaning tasks.

Q2. Code gathering tools can be picked up quickly and *readily applied to new use cases.*
Contributions encouraged: github.com/Microsoft/gather