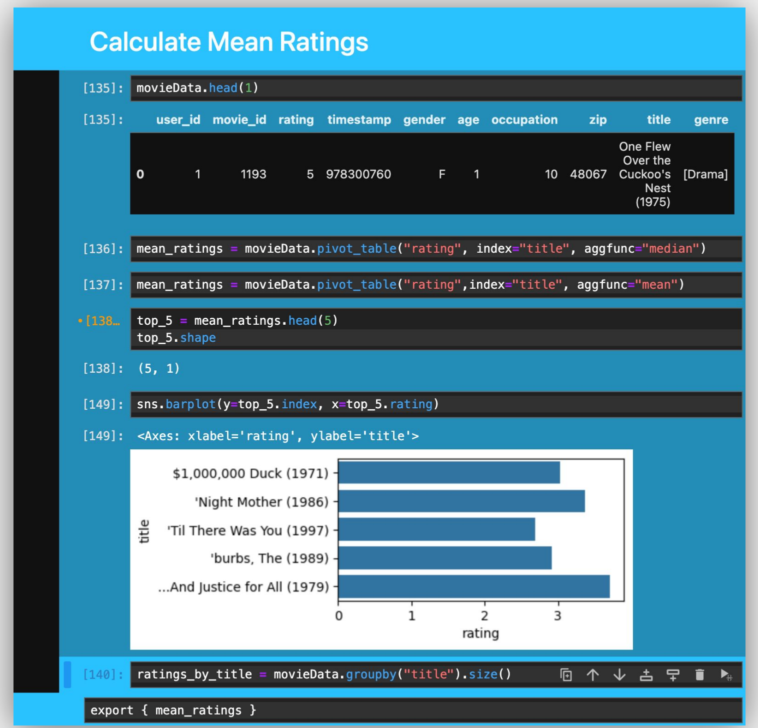
Pagebreaks

Multi-Cell Scopes in Computational Notebooks

Eric Rawn and Sarah Chasins
University of California, Berkeley



Calculate Mean Ratings

```
[135]: movieData.head(1)
```

[135]:

	user_id	movie_id	rating	timestamp	gender	age	occupation	zip	title	genre
0	1	1193	5	978300760	F	1	10	48067	One Flew Over the Cuckoo's Nest (1975)	[Drama]

```
[136]: mean_ratings = movieData.pivot_table("rating", index="title", aggfunc="median")

[137]: mean_ratings = movieData.pivot_table("rating",index="title", aggfunc="mean")

•[138…  top_5 = mean_ratings.head(5)
        top_5.shape

[138]: (5, 1)

[149]: sns.barplot(y=top_5.index, x=top_5.rating)

[149]: <Axes: xlabel='rating', ylabel='title'>
```

```
[140]: ratings_by_title = movieData.groupby("title").size()

export { mean_ratings }
```

Computational Notebooks are confusing.

Computational Notebooks are confusing. [1][2][3][4][5][6][7]

[1] Souti Chattopadhyay, Ishita Prasad, Austin Z. Henley, Anita Sarma, and Titus Barik. 2020. What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376729

[2] Taijara Loiola De Santana, Paulo Anselmo Da Mota Silveira Neto, Eduardo Santana De Almeida, and Iftekhar Ahmed. 2024. Bug Analysis in Jupyter Notebook Projects: An Empirical Study. ACM Trans. Softw. Eng. Methodol. 33, 4 (April 2024), 101:1–101:34. https://doi.org/10.1145/3641539

[3] Adam Rule, Amanda Birmingham, Cristal Zuniga, Ilkay Altintas, Shih-Cheng Huang, Rob Knight, Niema Moshiri, Mai H. Nguyen, Sara Brin Rosenthal, Fernando P.rez, and Peter W. Rose. 2019. Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. PLOS Computational Biology 15, 7 (July 2019), e1007007. https://doi.org/10.1371/journal.pcbi.1007007 Publisher: Public Library of Science.

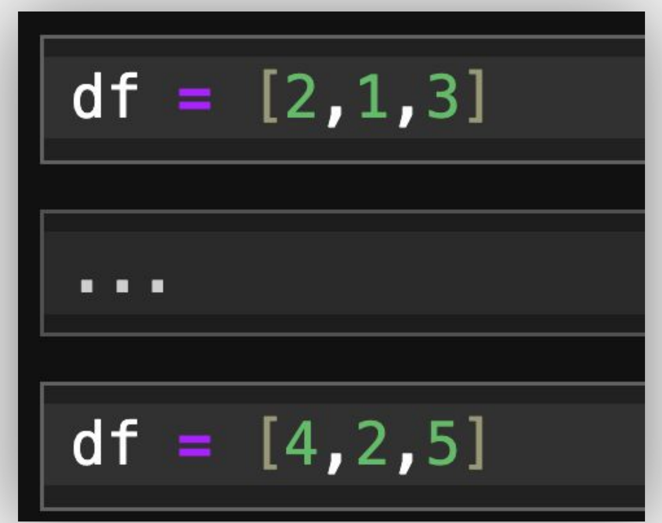
[4] Jeremy Singer. 2020. Notes on notebooks: is Jupyter the bringer of jollity?. In Proceedings of the 2020 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward! 2020). Association for Computing Machinery, New York, NY, USA, 180–186. https://doi.org/10.1145/3426428.3426924

[5] April Yi Wang, Anant Mittal, Christopher Brooks, and Steve Oney. 2019. How Data Scientists Use Computational Notebooks for Real-Time Collaboration. Proc. ACM Hum.-Comput. Interact. 3, CSCW (Nov. 2019), 39:1–39:30. https://doi.org/10.1145/3359141

[6] Nathaniel Weinman, Steven M. Drucker, Titus Barik, and Robert DeLine. 2021. Fork It: Supporting Stateful Alternatives in Computational Notebooks. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21). Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3411764.3445527

[7] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E. John, and Brad A. Myers. 2018. The Story in the Notebook: Exploratory Data Science using a Literate Programming Tool. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, Montreal QC Canada, 1–11. https://doi.org/10.1145/3173574.3173748

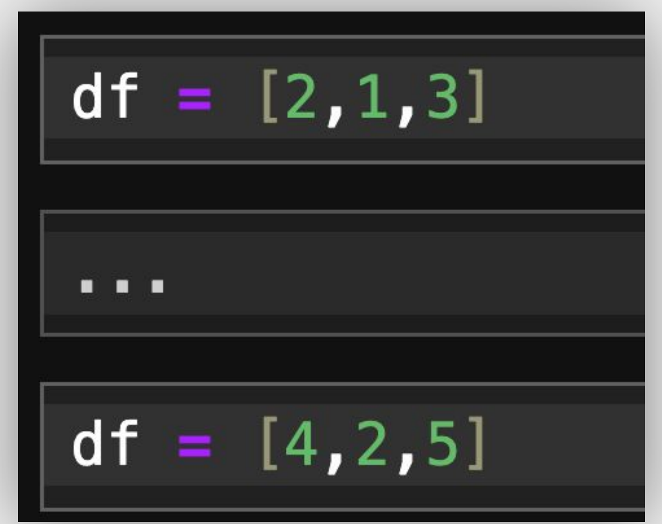
Global Variables in Computational Notebooks are confusing.



```
df = [2,1,3]
```

```
...
```

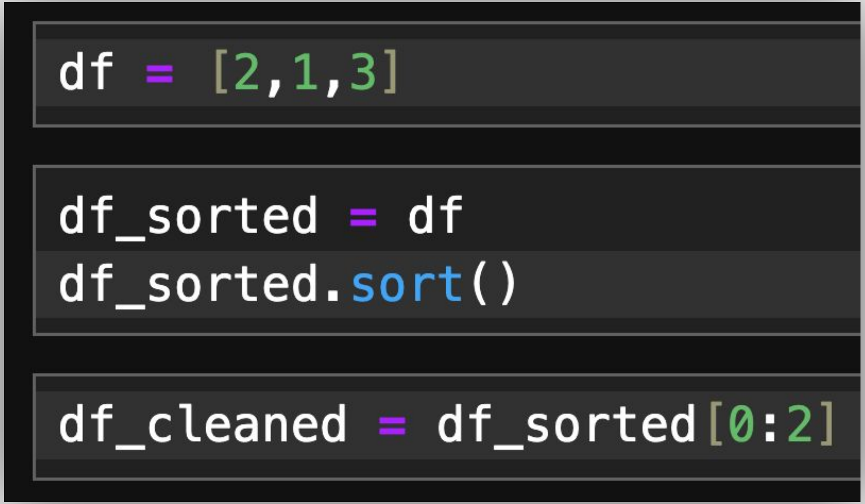
```
df = [4,2,5]
```



```
df = [2,1,3]
```

```
...
```

```
df = [4,2,5]
```



```
df = [2,1,3]
```

```
df_sorted = df
df_sorted.sort()
```

```
df_cleaned = df_sorted[0:2]
```

But Eric, shouldn't they just use functions?

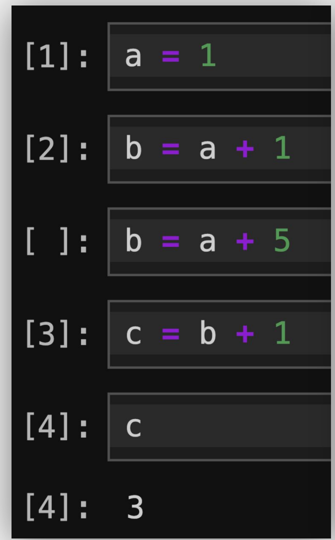
But Eric, shouldn't they just use functions?[1][2][3][4][5]

[1] Taijara Loiola De Santana, Paulo Anselmo Da Mota Silveira Neto, Eduardo Santana De Almeida, and Iftekhar Ahmed. 2024. Bug Analysis in Jupyter Notebook Projects: An Empirical Study. ACM Trans. Softw. Eng. Methodol. 33, 4 (April 2024), 101:1–101:34. https://doi.org/10.1145/3641539
[2] Adam Rule, Amanda Birmingham, Cristal Zuniga, Ilkay Altintas, Shih-Cheng Huang, Rob Knight, Niema Moshiri, Mai H. Nguyen, Sara Brin Rosenthal, Fernando P.rez, and Peter W. Rose. 2019. Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. PLOS Computational Biology 15, 7 (July 2019), e1007007. https://doi.org/10.1371/journal.pcbi.1007007 Publisher: Public Library of Science.
[3] Jeremy Singer. 2020. Notes on notebooks: is Jupyter the bringer of jollity?. In Proceedings of the 2020 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward! 2020). Association for Computing Machinery, New York, NY, USA, 180–186. https://doi.org/10.1145/3426428.3426924
[4]Helen Dong, Shurui Zhou, Jin L.C. Guo, and Christian Kästner. 2021. Splitting, Renaming, Removing: A Study of Common Cleaning Activities in Jupyter Notebooks. In 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW). 114–119. https: //doi.org/10.1109/ASEW52652.2021.00032 ISSN: 2151-0830.
[5] Luigi Quaranta, Fabio Calefato, and Filippo Lanubile. 2022. Eliciting Best Practices for Collaboration with Computational Notebooks. Proc. ACM Hum.-Comput. Interact. 6, CSCW1 (April 2022), 87:1–87:41. https://doi.org/10.1145/3512934

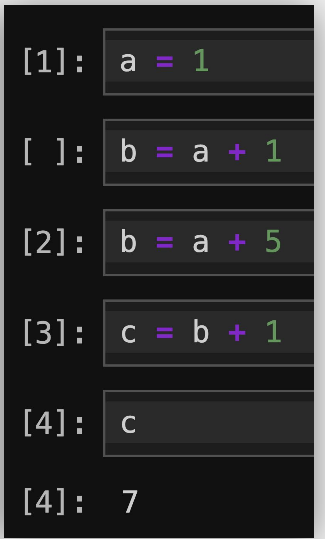
Functions interfere with exploratory interactions

Exploratory Interactions in Notebook Programming (Kery 2022)

Cells as Impromptu Versions

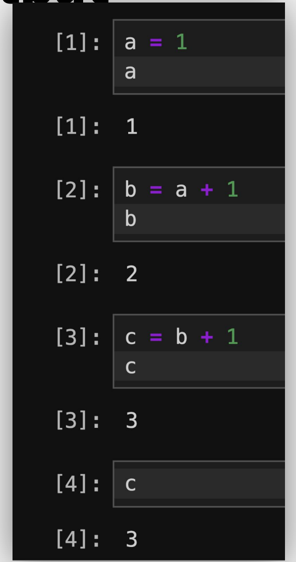


```
[1]: a = 1

[2]: b = a + 1

[ ]: b = a + 5

[3]: c = b + 1

[4]: c

[4]: 3
```



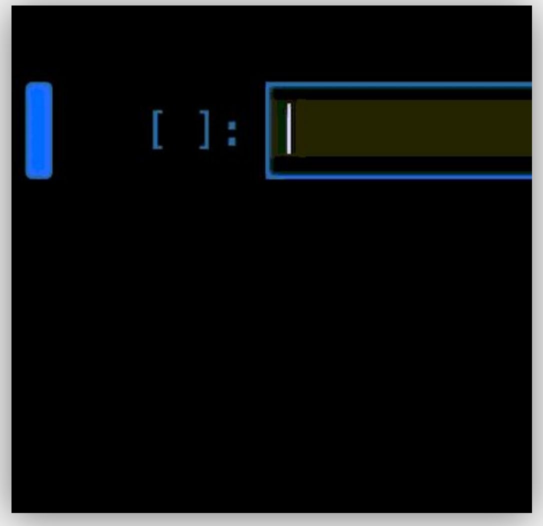
```
[1]: a = 1

[ ]: b = a + 1

[2]: b = a + 5

[3]: c = b + 1

[4]: c

[4]: 7
```

Interleaving Code and Output



```
[1]: a = 1
     a

[1]: 1

[2]: b = a + 1
     b

[2]: 2

[3]: c = b + 1
     c

[3]: 3

[4]: c

[4]: 3
```

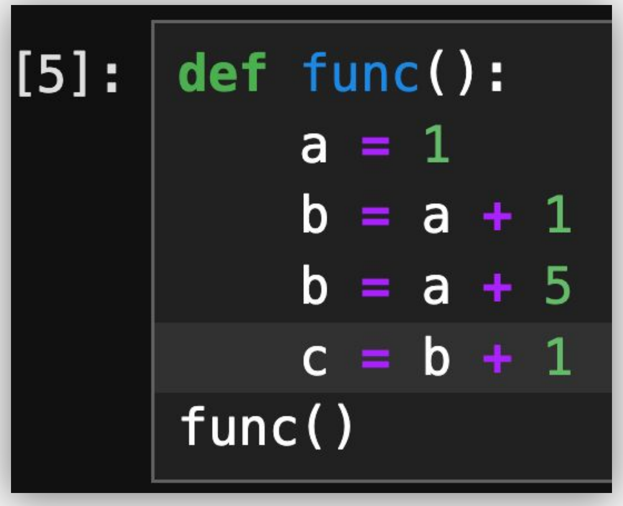
Iterative Development



```
[ ]:
```

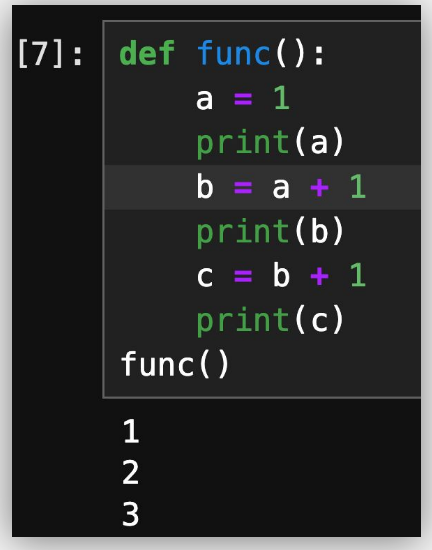
Functions interfere with exploratory interactions

Functions must be defined in a single cell



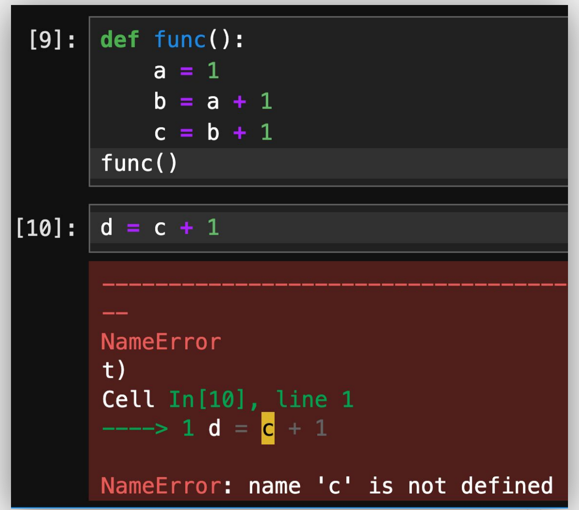
```
[5]: def func():
         a = 1
         b = a + 1
         b = a + 5
         c = b + 1
     func()
```

Functions Output at the call site



```
[7]: def func():
         a = 1
         print(a)
         b = a + 1
         print(b)
         c = b + 1
         print(c)
     func()

     1
     2
     3
```

Functions scope their variables in a single cell



```
[9]: def func():
         a = 1
         b = a + 1
         c = b + 1
     func()

[10]: d = c + 1

     ---------------------------------------
     --
     NameError
     t)
     Cell In[10], line 1
     ----> 1 d = c + 1

     NameError: name 'c' is not defined
```

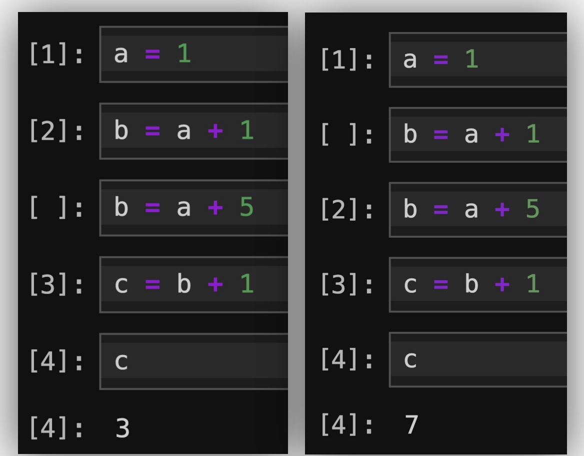
How can we help mitigate the confusions with global variables *without interfering with exploratory interactions*?

Functions….

- Name a chunk of code
- Give a way to call that chunk multiple times
- Allow named parameters
- Provide new control flow options
- Introduce a scope

1. **Sharing State Implicitly Between Neighboring Cells**
2. **Manual Execution of Cells**

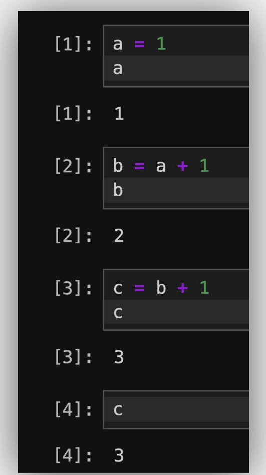
Cells as Impromptu Versions



```
[1]:  a = 1
[2]:  b = a + 1
[ ]:  b = a + 5
[3]:  c = b + 1
[4]:  c
[4]:  3
```

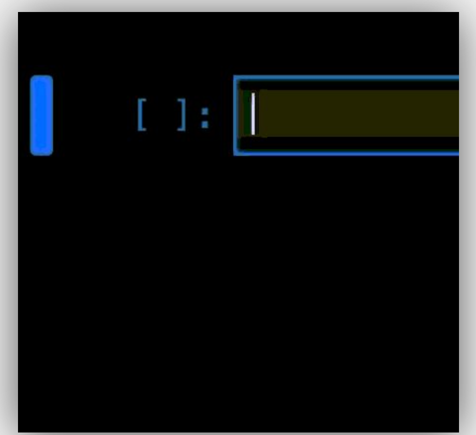
```
[1]:  a = 1
[ ]:  b = a + 1
[2]:  b = a + 5
[3]:  c = b + 1
[4]:  c
[4]:  7
```

Interleaving Code and Output



```
[1]:  a = 1
      a
[1]:  1
[2]:  b = a + 1
      b
[2]:  2
[3]:  c = b + 1
      c
[3]:  3
[4]:  c
[4]:  3
```

Iterative Development

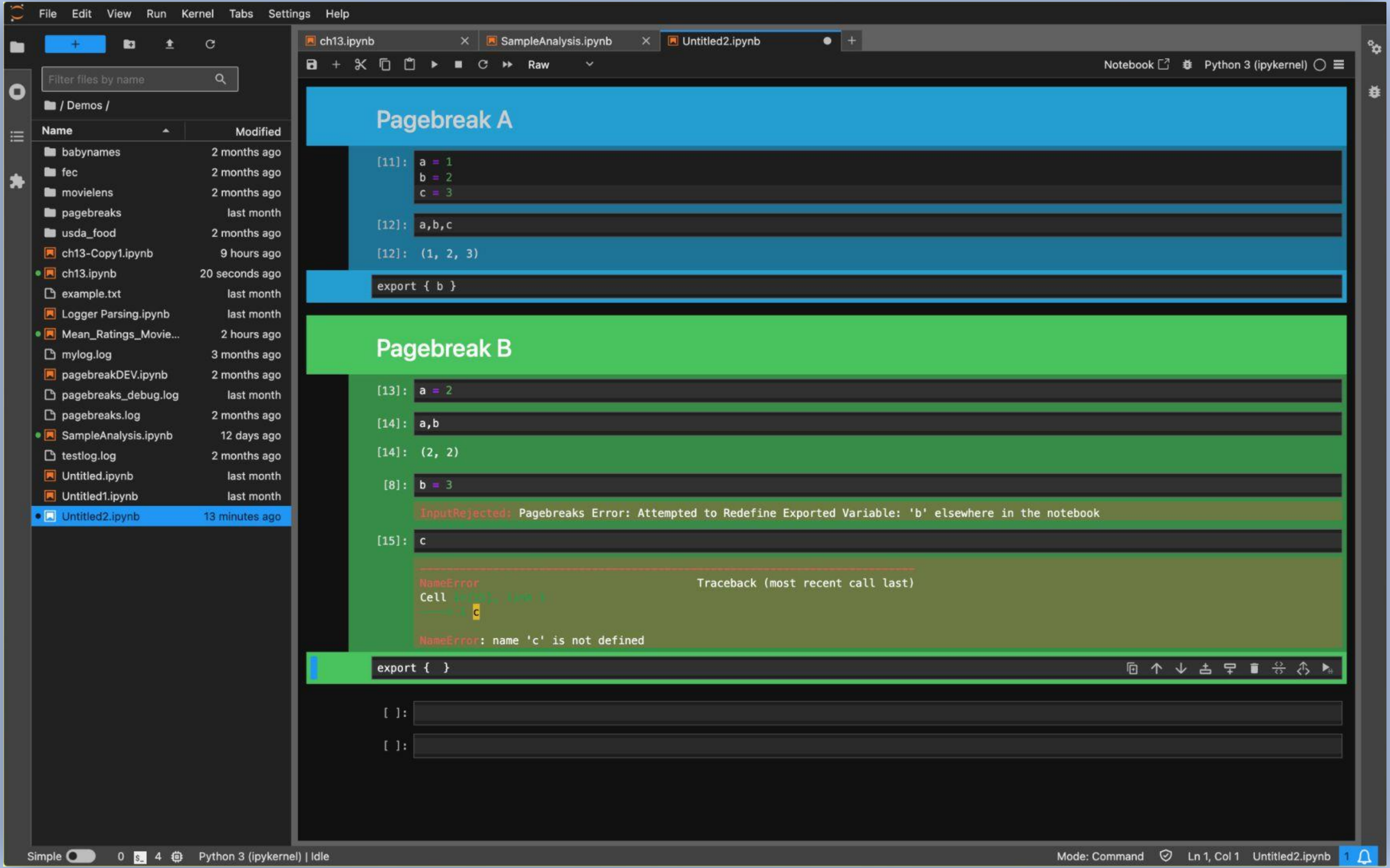


```
[ ]:
```

our new construct should give programmers a way to ...

(1) communicate between cells without writing into global state

(2) independently execute an arbitrary number of communicating cells



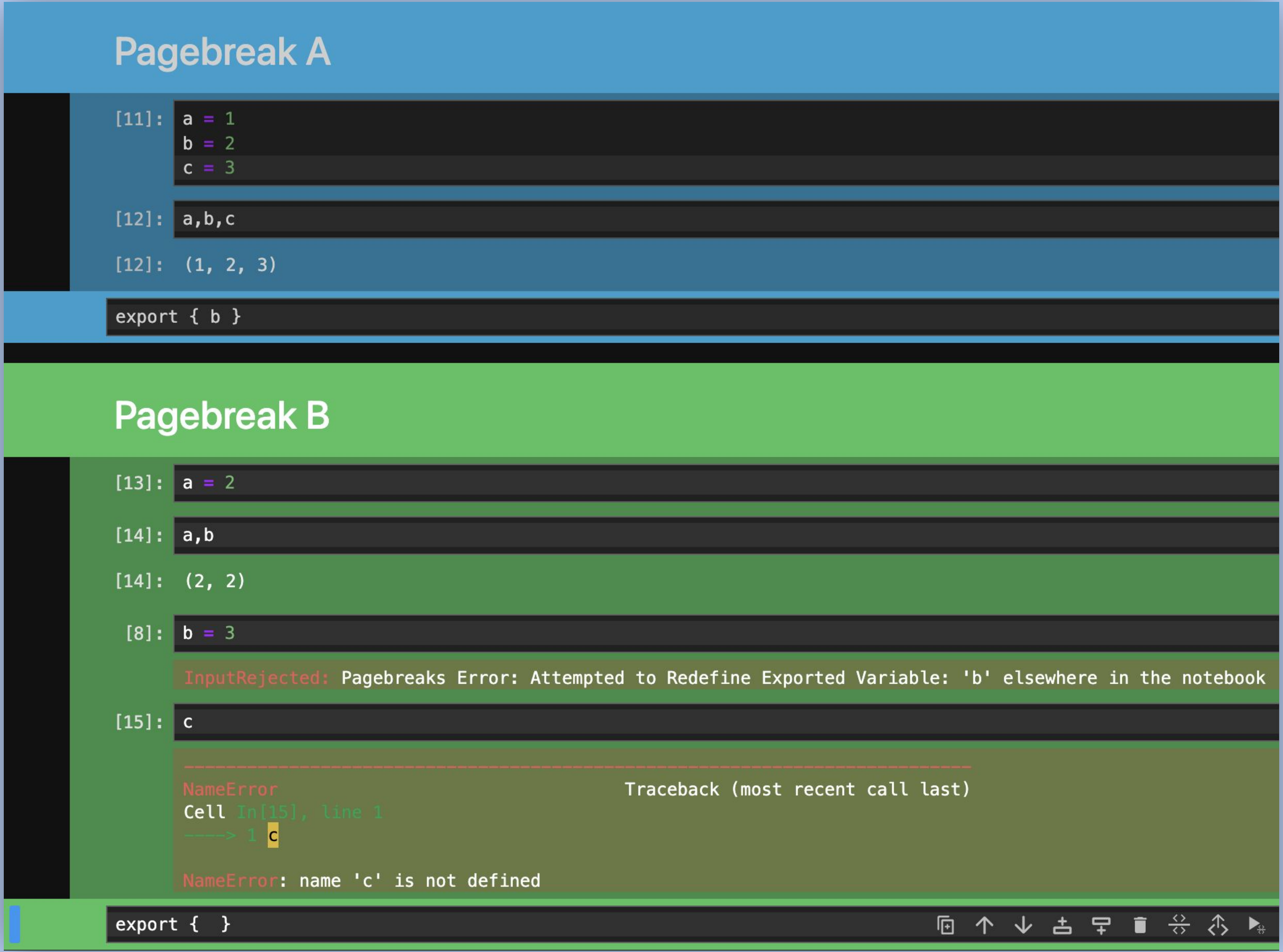
Name	Modified
babynames	2 months ago
fec	2 months ago
movielens	2 months ago
pagebreaks	last month
usda_food	2 months ago
ch13-Copy1.ipynb	9 hours ago
ch13.ipynb	20 seconds ago
example.txt	last month
Logger Parsing.ipynb	last month
Mean_Ratings_Movie...	2 hours ago
mylog.log	3 months ago
pagebreakDEV.ipynb	2 months ago
pagebreaks_debug.log	last month
pagebreaks.log	2 months ago
SampleAnalysis.ipynb	12 days ago
testlog.log	2 months ago
Untitled.ipynb	last month
Untitled1.ipynb	last month
Untitled2.ipynb	13 minutes ago

Pagebreak A

```
[11]: a = 1
      b = 2
      c = 3

[12]: a,b,c

[12]: (1, 2, 3)

export { b }
```

Pagebreak B

```
[13]: a = 2

[14]: a,b

[14]: (2, 2)

 [8]: b = 3

InputRejected: Pagebreaks Error: Attempted to Redefine Exported Variable: 'b' elsewhere in the notebook

[15]: c

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[15], line 1
----> 1 c

NameError: name 'c' is not defined

export {  }

[ ]:

[ ]:
```

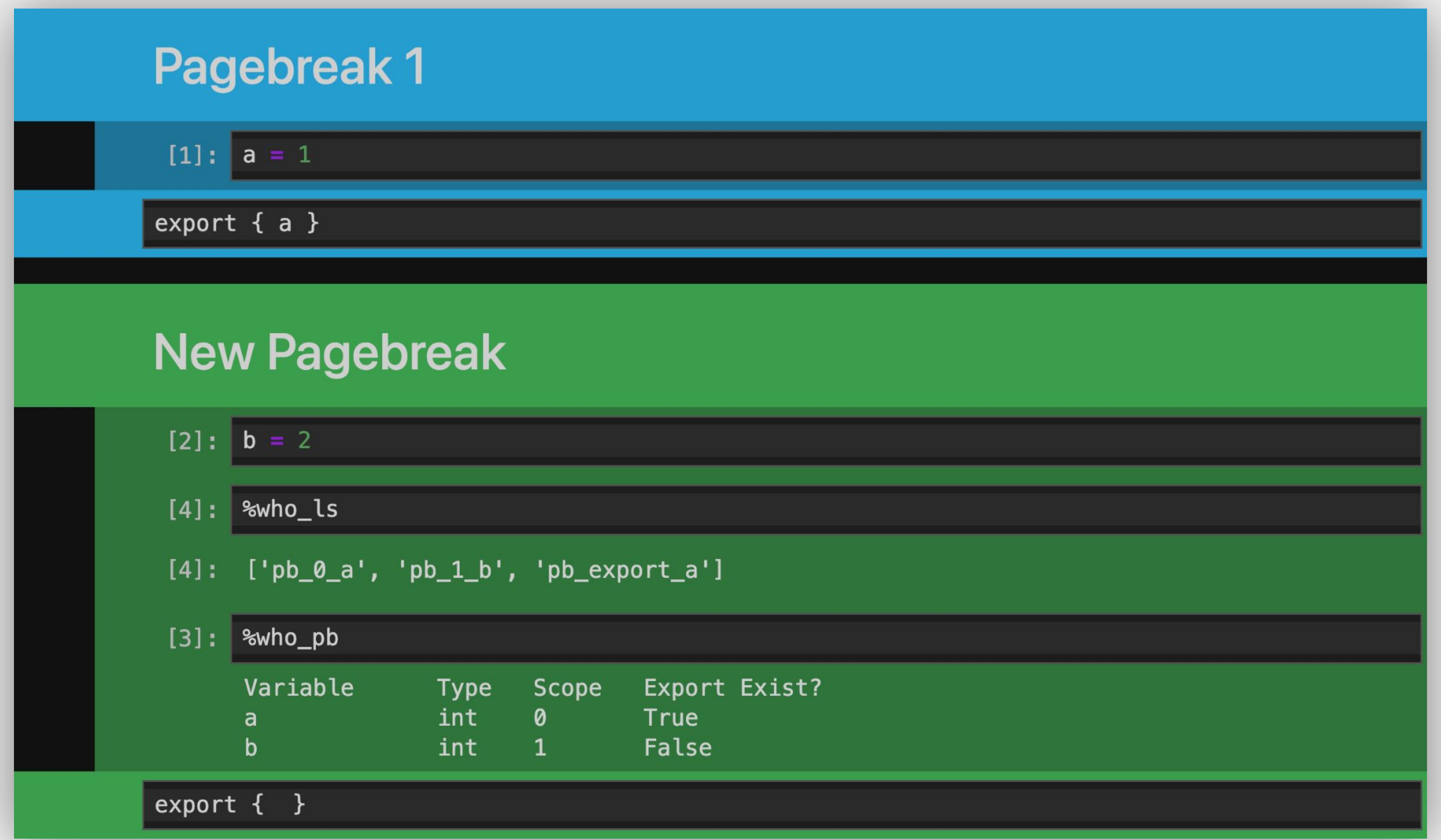


Pagebreak A

```
[11]: a = 1
      b = 2
      c = 3

[12]: a,b,c

[12]: (1, 2, 3)

export { b }
```

Pagebreak B

```
[13]: a = 2

[14]: a,b

[14]: (2, 2)

 [8]: b = 3

      InputRejected: Pagebreaks Error: Attempted to Redefine Exported Variable: 'b' elsewhere in the notebook

[15]: c

      ---------------------------------------------------------------------------
      NameError                                 Traceback (most recent call last)
      Cell In[15], line 1
      ----> 1 c

      NameError: name 'c' is not defined

export {  }
```



Pagebreak 1

```
[1]: a = 1
```

```
export { a }
```

New Pagebreak

```
[2]: b = 2
```

```
[4]: %who_ls
```

```
[4]: ['pb_0_a', 'pb_1_b', 'pb_export_a']
```

```
[3]: %who_pb
```

```
Variable      Type   Scope   Export Exist?
a             int    0       True
b             int    1       False
```

```
export {  }
```

What happened when we gave Pagebreaks to notebook programmers?

What happened when we gave Pagebreaks to notebook programmers?

Study Design:

- 5 Participants
- ~2 week usage in their own work
- 1-1.5 hour interview at the end

- Pagebreaks helped address issues with global variables

- It didn’t seem to interfere with exploratory interactions

"To me the benefit of Jupyter notebooks is that you can run things line-by-line in cells ... and I think of Pagebreaks doing that but to a higher level (P3)".

Participants used Pagebreaks to organize their notebooks

Participants used Pagebreaks to organize their notebooks

- Compartmentalization

"It gives me [a] guarantee that I know that what I've run is what is meant to be running (P4)".

Participants used Pagebreaks to **organize** their notebooks

- Compartmentalization
- **Dataflow**

"it makes me think more about the namespace of variables ... [makes me] more intentional about ... [what] I want to export from one Pagebreak to another ... [and] the things I want to reuse (P4)."

Participants used Pagebreaks to organize their notebooks

- Compartmentalization
- Dataflow
- **By-Purpose**

In Defense of "Messy" Programming

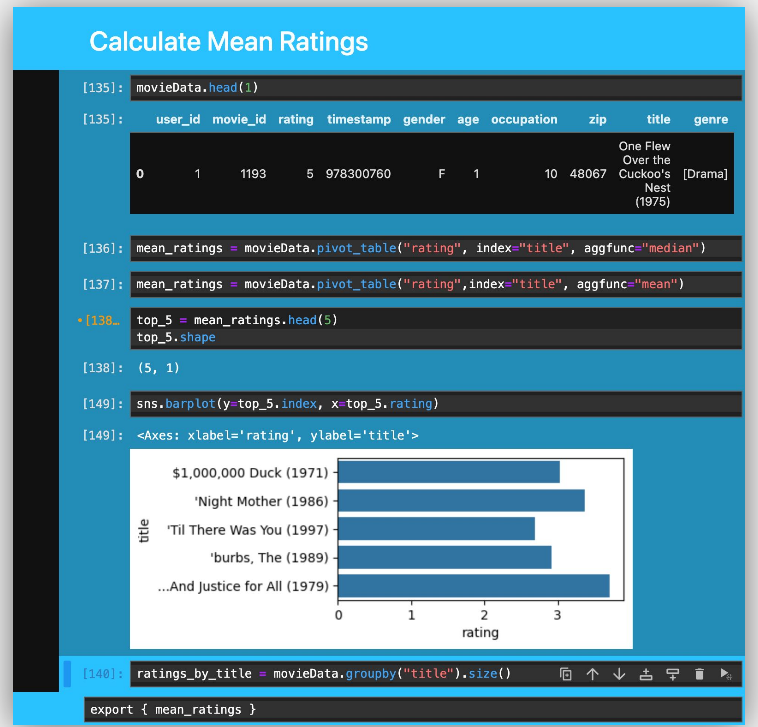
Breaking Open Functions

Designing for Programming Languages and Programming Environments Together

Pagebreaks

Multi-Cell Scopes in Computational Notebooks

Eric Rawn and Sarah Chasins
University of California, Berkeley



Calculate Mean Ratings

```
[135]: movieData.head(1)
```

[135]:

	user_id	movie_id	rating	timestamp	gender	age	occupation	zip	title	genre
0	1	1193	5	978300760	F	1	10	48067	One Flew Over the Cuckoo's Nest (1975)	[Drama]

```
[136]: mean_ratings = movieData.pivot_table("rating", index="title", aggfunc="median")

[137]: mean_ratings = movieData.pivot_table("rating",index="title", aggfunc="mean")

•[138…  top_5 = mean_ratings.head(5)
        top_5.shape

[138]: (5, 1)

[149]: sns.barplot(y=top_5.index, x=top_5.rating)

[149]: <Axes: xlabel='rating', ylabel='title'>
```

```
[140]: ratings_by_title = movieData.groupby("title").size()

export { mean_ratings }
```