Full paper & more!

willepperson.com/papers/reuse-ds

Strategies for Reuse and Sharing among Data Scientists in Software Teams

ICSE SEIP ‘22

Will Epperson
Carnegie Mellon University

April Yi Wang
The University of Michigan

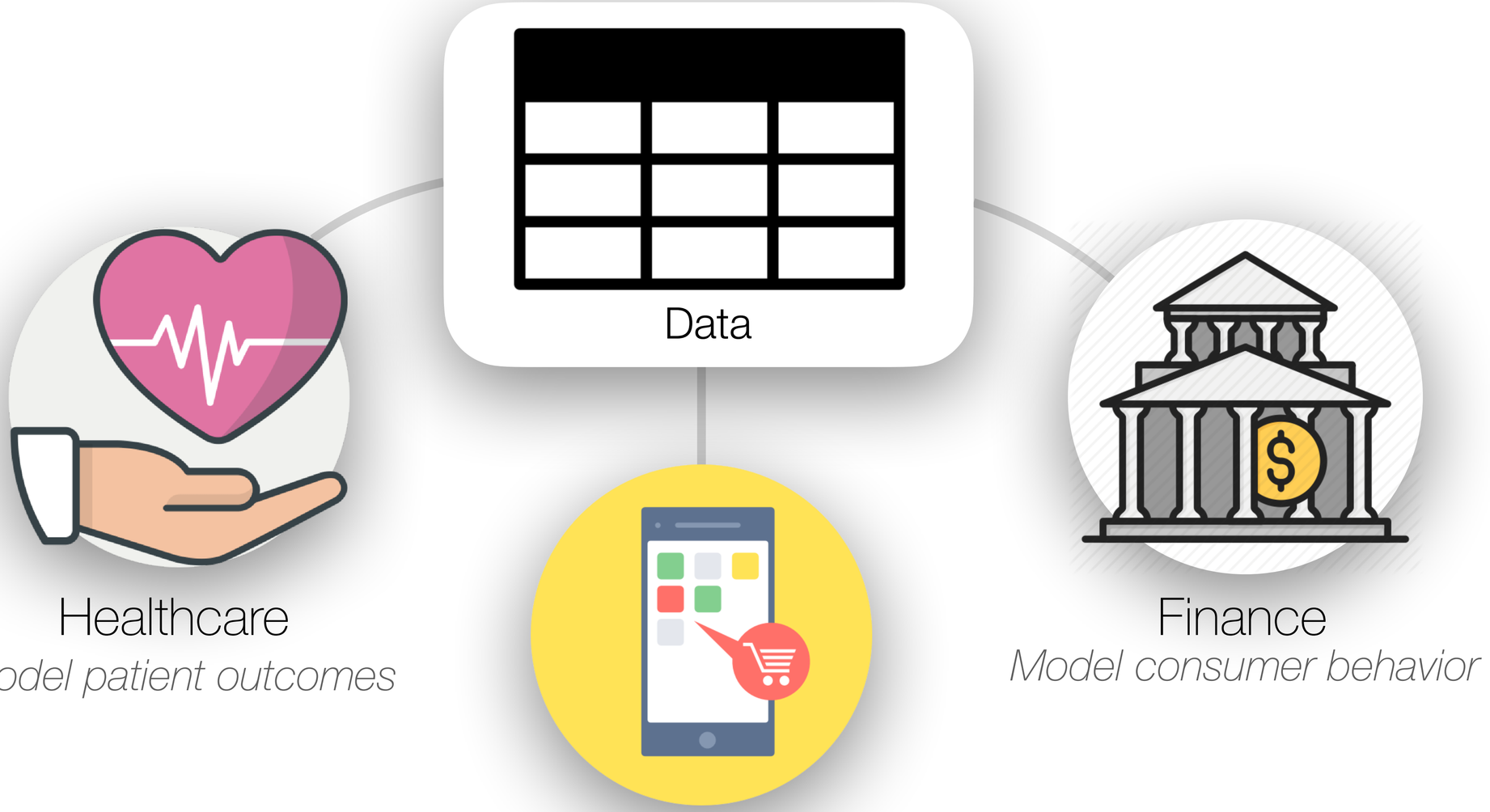
Robert DeLine
Microsoft Research

Steven Drucker
Microsoft Research

Data Interaction Group

University of Michigan

Microsoft


Data
Healthcare
Model patient outcomes
Technology
Understand user behavior
Finance
Model consumer behavior

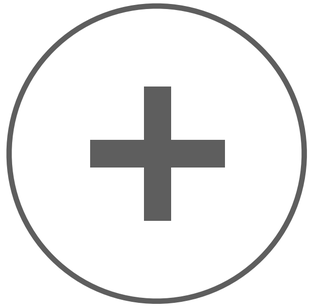
Data Engineer

Software Engineer

Data Scientist

Business Analyst

Data Analyst



More…

No unicorn data scientists that can do it all!

Davenport, T. Beyond Unicorns: Educating, Classifying, and Certifying Business Data Scientists. Harvard Data Science Review, 2(2). 2020.

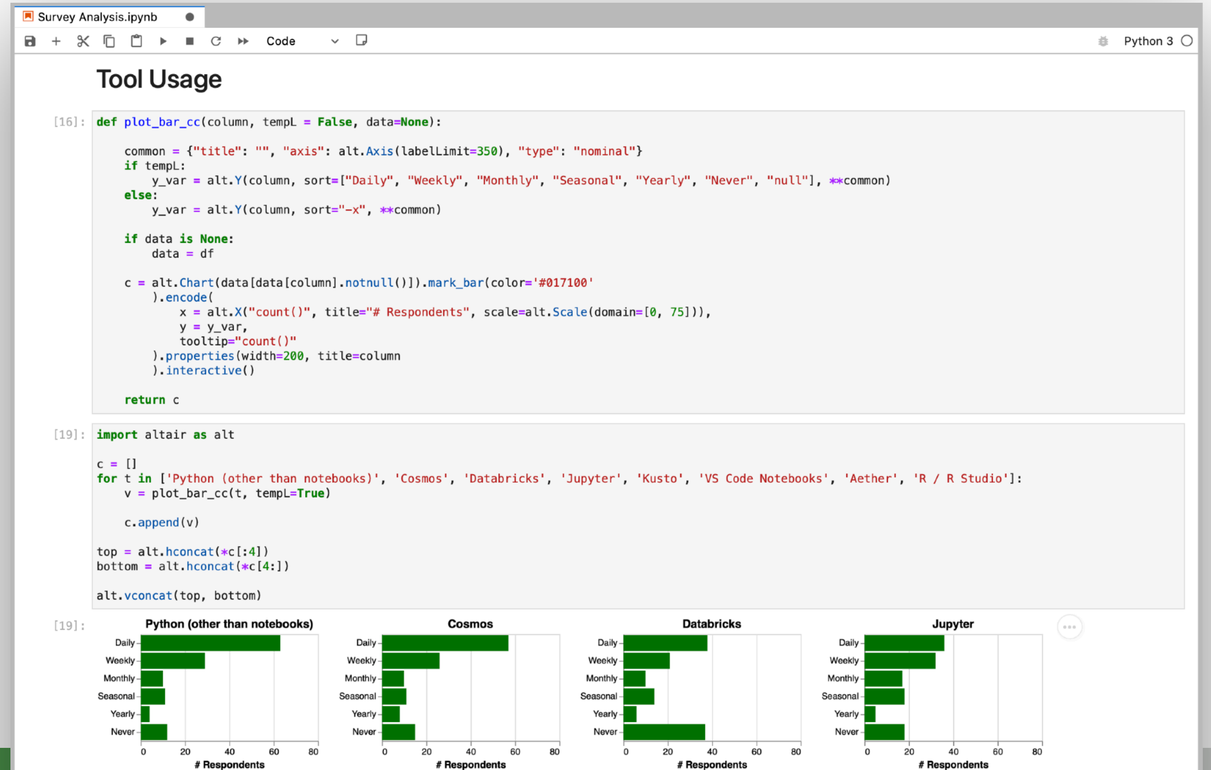
Data science is highly iterative, exploratory

Data Scientist

Computational notebooks

…have some issues:

1. Messy, non-reproducible
2. Lead to bad coding practices because of unexpected execution order and lack of modular code
3. Users tend to think less about future use of their code



Amy X. Zhang, Michael J. Muller, and Dakuo Wang. How do Data Science Workers Collaborate? Roles, Workflows, and Tools. 2020.

Kim, Miryung et al. Data Scientists in Software Teams: State of the Art and Challenges. IEEE Transactions on Software Engineering 44,11. 2018.

Adam Rule, et al. Exploration and Explanation in Computational Notebooks. 2018.

Reuse is the standard in Software Engineering

Modular, reusable code has been staple of effective software engineering

Rich online repositories help facilitate reuse and sharing

Proper reuse leads to fewer bugs, higher productivity

Principles like “DRY”: Don't Repeat Yourself

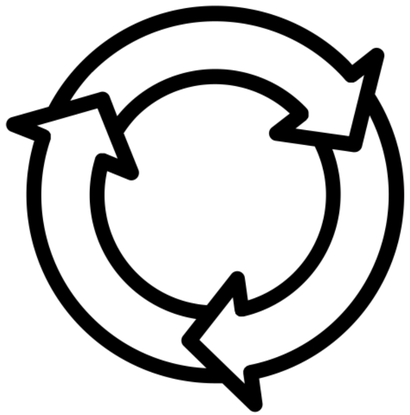
Parastoo Mohagheghi and Reidar Conradi. Quality, productivity and economic benefits of software reuse: a review of industrial studies. 2007

William B Frakes and Kyo Kang. Software reuse research: Status and future. 2005.

Research Question

How does this exploratory practice of data science fit with the fundamental task of finding, reusing, and sharing past analysis work?

Finding past analysis code



Reusing own or others' past analysis code

Sharing analyses between data scientists

Methods

Interview Study

17 participants

Data scientists or equivalent job titles at Microsoft

1 hr interviews

Describe their experience with reusing past analysis work

Generated themes on reuse and sharing

Follow Up Survey

132 respondents

Data scientists or equivalent job titles at Microsoft (same pool)

10-15 min survey

Assess prevalence of strategies, themes identified in interviews

Who did we interview?

In both studies, came from wide variety of teams, backgrounds, experience

Experience ranging from 3 months to 35 years

Education:

- Wide variety; undergrad degrees to PhD, technical fields to humanities backgrounds
- No common training on how to do data science…

Small teams

Types of teams:

- 1-22 people (mean of 8)
- Centralized data science team
- Or Data Scientist embedded on product / software teams

On most projects, data scientists are working **alone or in pairs** to do analysis

Generating themes

Used open coding to generate 97 different codes

Two of the authors worked to apply these codes to the transcripts

Data Science Workflows

5 strategies for reuse and sharing

Influencers on sharing practices

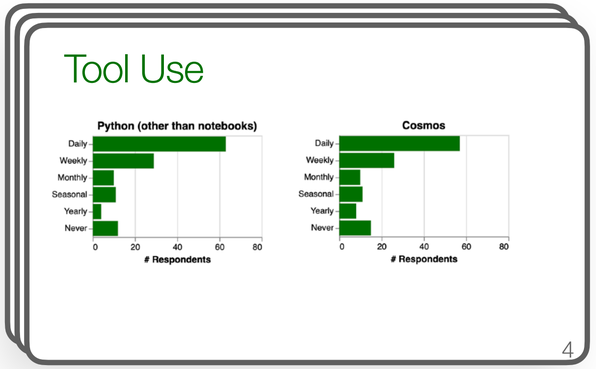
Data Science Workflows

Outcome of data science is not the code

Data storage → tool usage → sharing and re-use practices

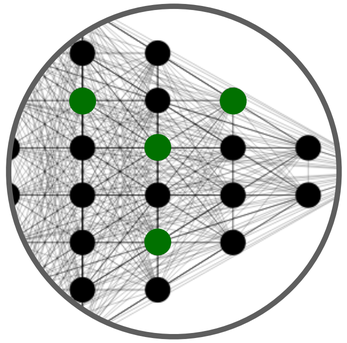
Outcome of data science is not analysis code

Instead it is…



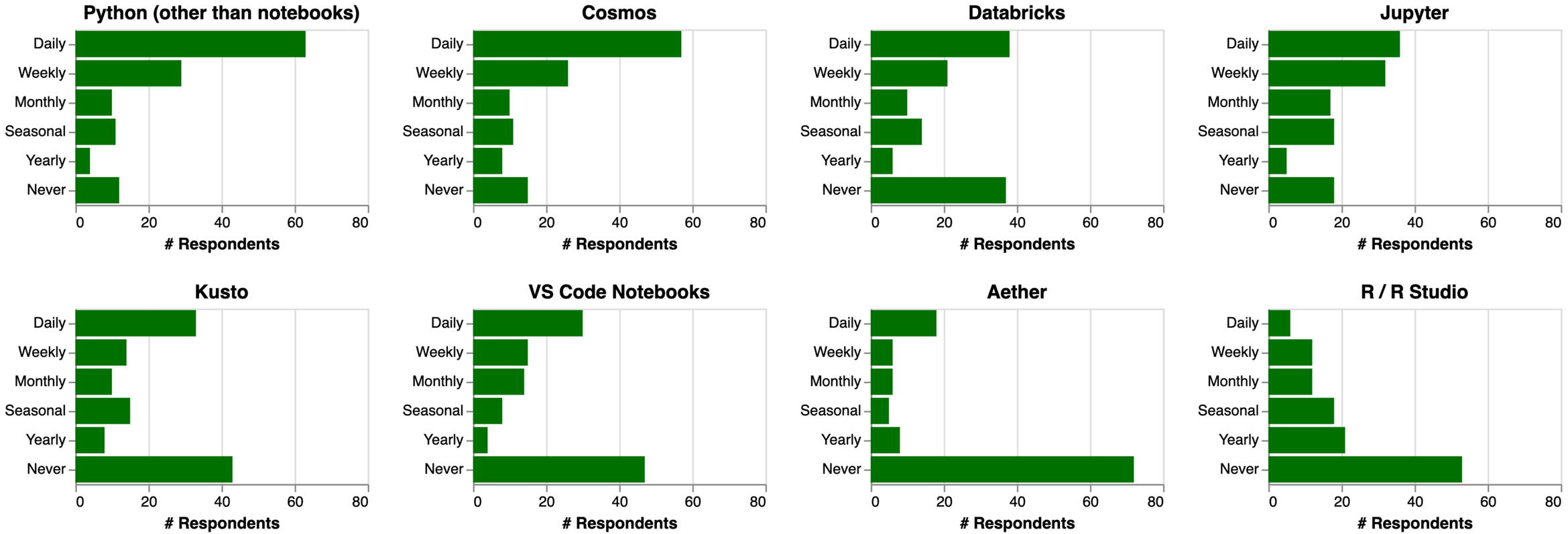
Slide decks

Reports



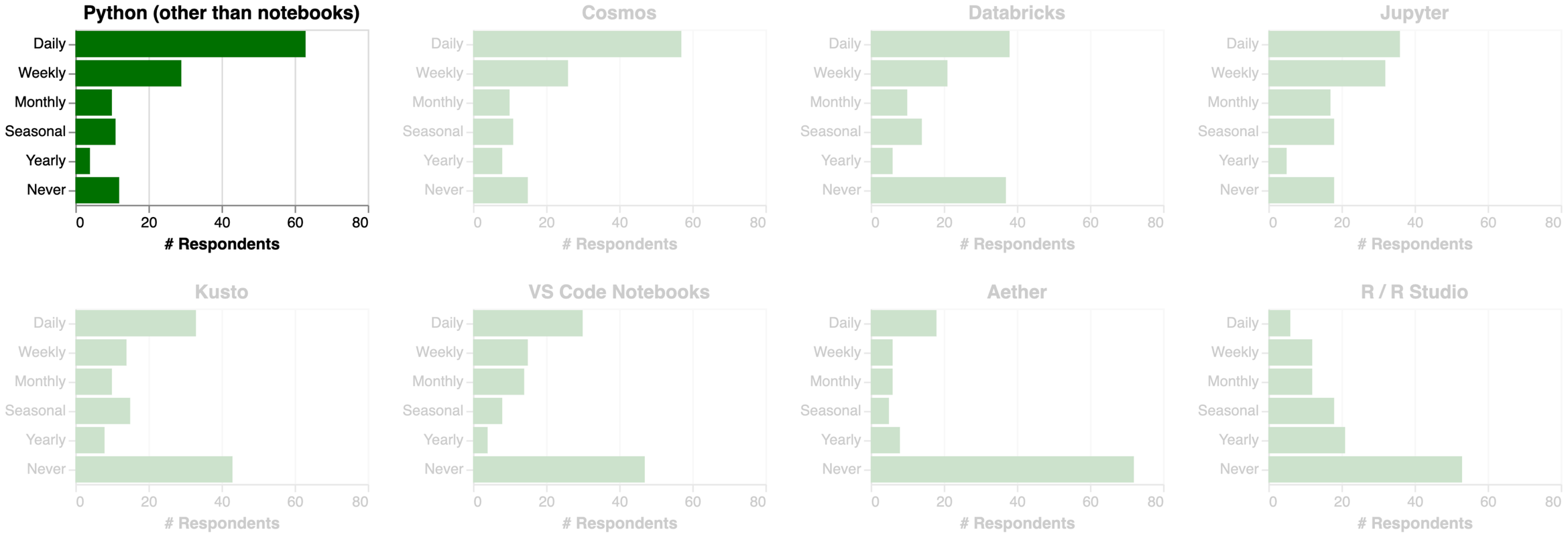
Models

Tools shape sharing and reuse



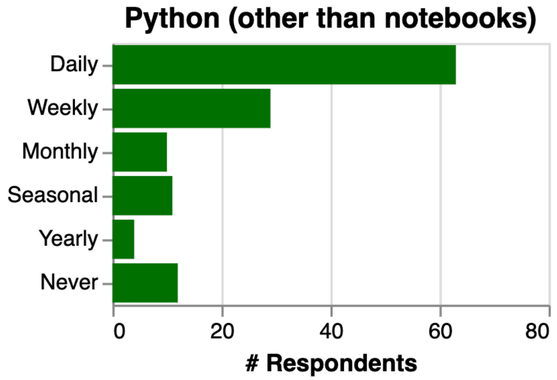
Most Commonly Used Tools
(top left to bottom right)

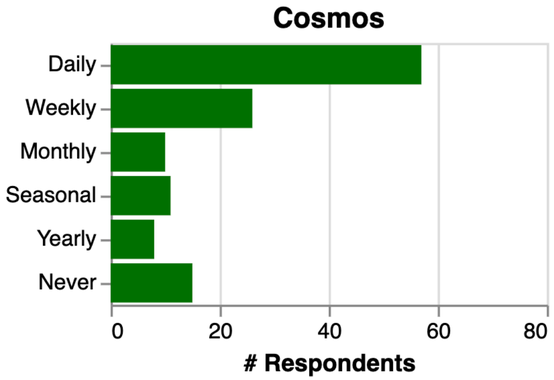
Tools shape sharing and reuse

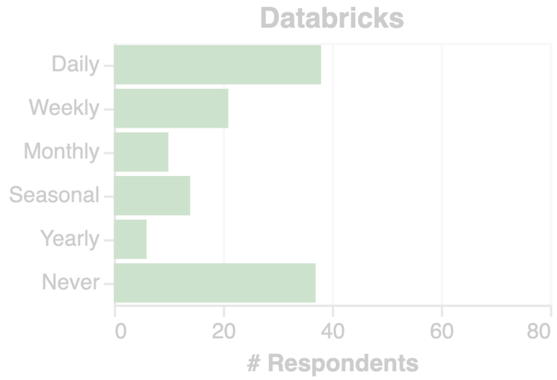


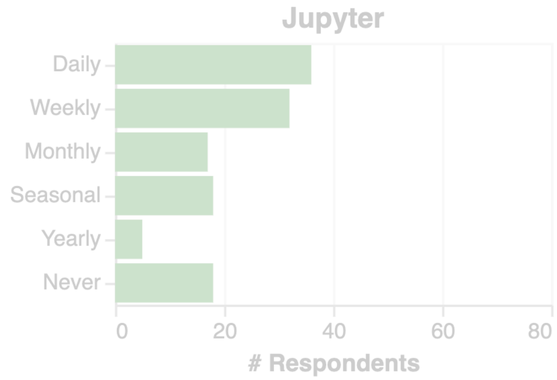
Most Commonly Used Tools
(top left to bottom right)

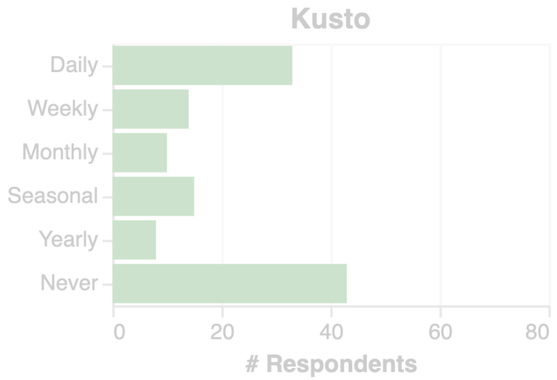
Tools shape sharing and reuse

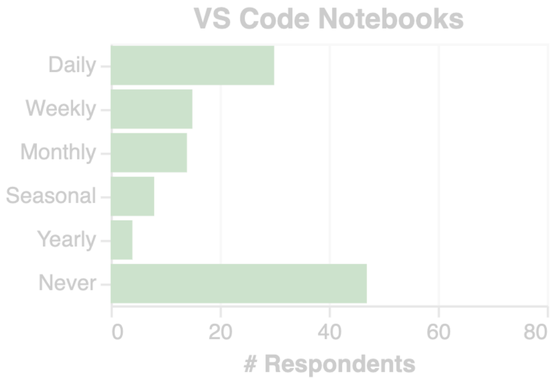


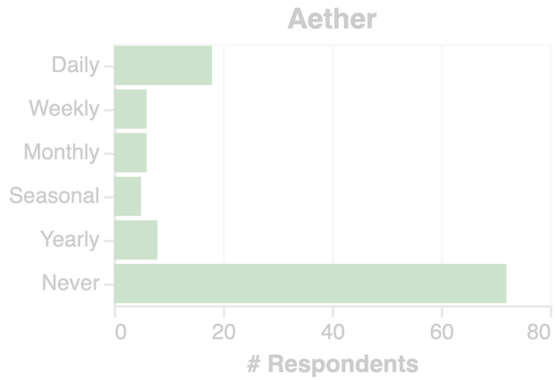


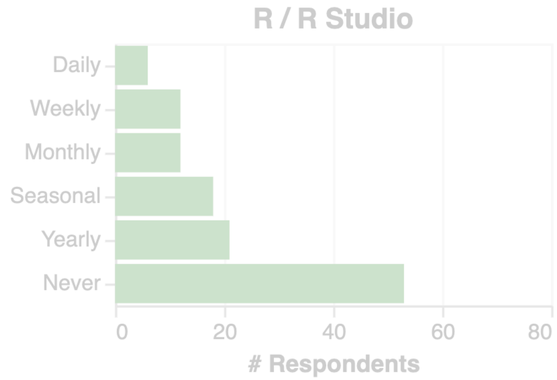






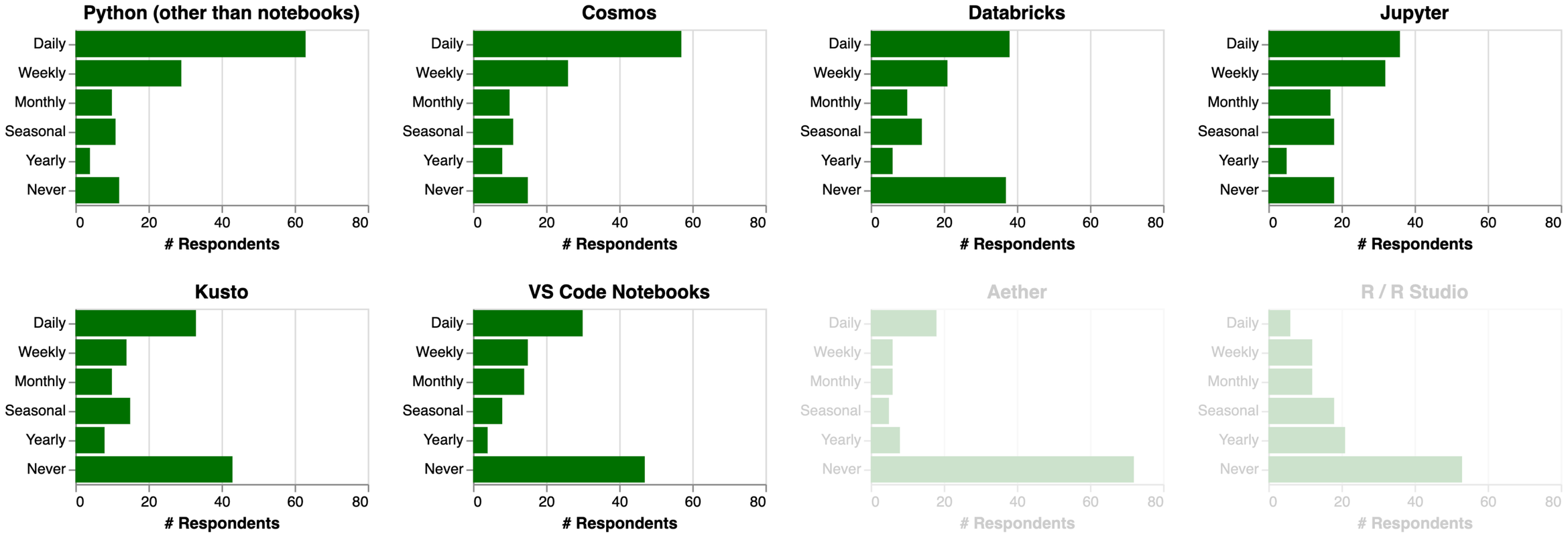






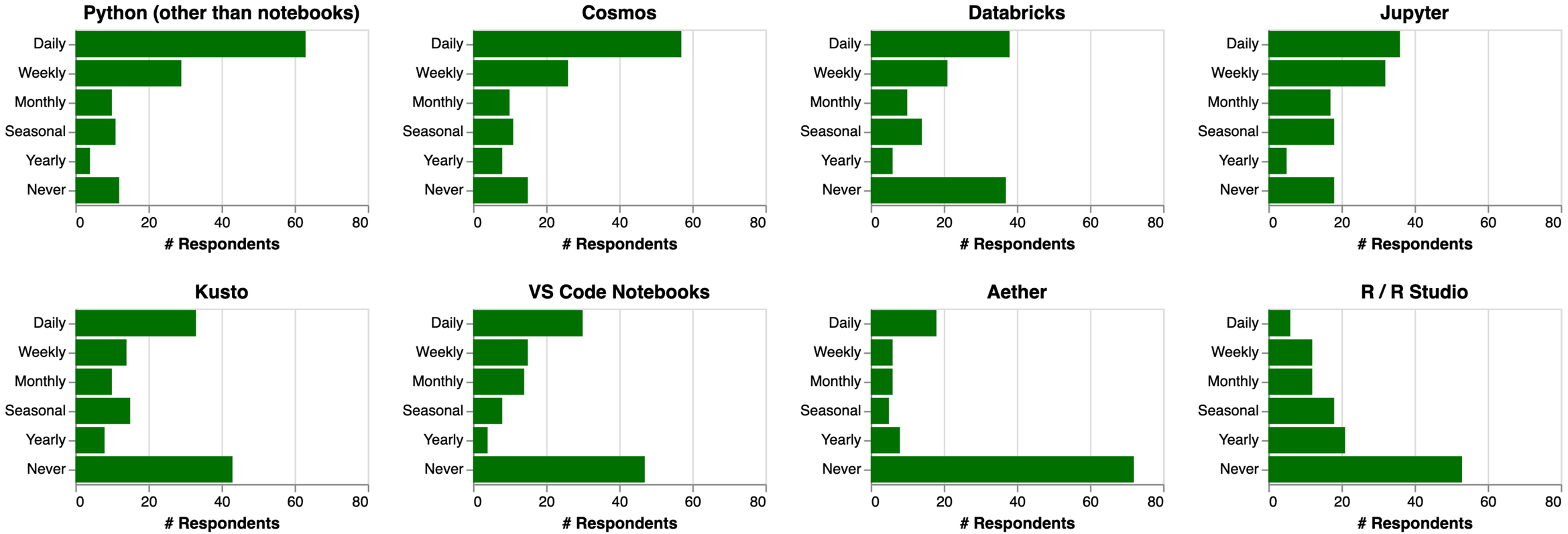
Most Commonly Used Tools
(top left to bottom right)

Tools shape sharing and reuse



Most Commonly Used Tools
(top left to bottom right)

Tools shape sharing and reuse



Most Commonly Used Tools
(top left to bottom right)

Tools shape sharing and reuse

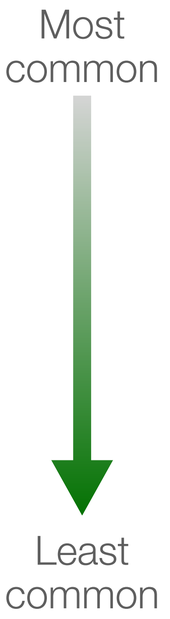
Wide variety of tools and languages used for analysis that are adapted to be **reusable** based on affordances of the platform!

Most Commonly Used Tools
(top left to bottom right)

5 Strategies for Reuse and Sharing

What functionality?

Did not vary much across strategies



1. Data pre-processing, transformation, cleaning
2. Reading and writing data
3. Modeling and evaluation code
4. Data visualization and reporting
5. Other tasks (model ensembling, selection, post-processing)

1. **Personal** Analysis Reuse
2. **Personal** Utility Libraries
3. **Team** Shared Analysis Code
4. **Team** Shared Template Notebooks
5. **Team** Shared Libraries

1 **Personal** Analysis Reuse

Finding own past work and reusing.

- ❖ “Anything I know I did before, I reuse it.” (SP63)
- ❖ Not “cleaned” before storage

Easy, quick

But… no formal versioning leads to “analysis-v1”, “analysis-v2”

2 **Personal** Utility Libraries

"These files are just calls which I have been using for three or five years now. I just constantly go to them again and again and again. So, **I extracted these very, very generic common things in single repo called utils** that each file just does a single thing." - IP12

Short snippets of code that are often reused; worth the time investment to refactor into reusable function

Distinct from 1 **Personal** Analysis Reuse because requires **intentional cleaning** of past analysis code

3 **Team** Shared Analysis Code

Central repo for shared code among a **team**

- ❖ On GitHub, or more ad hoc (over messaging or email)
- ❖ Sharing both computational notebooks and scripts
- ❖ Code is **cleaned** before sharing

3 **Team** Shared Analysis Code

Avoid repeated work! No ability to reuse others' work **leads to frustration**:

"Oh yes, tons of work was repeated in many places. Because if you weren't a part of the team that had access to some shared drive somewhere, then **you didn't know that [work] existed.**" — IP16

3 **Team** Shared Analysis Code

Requires a **time investment to clean code**

Tension between **using** shared code and **contributing** new code:

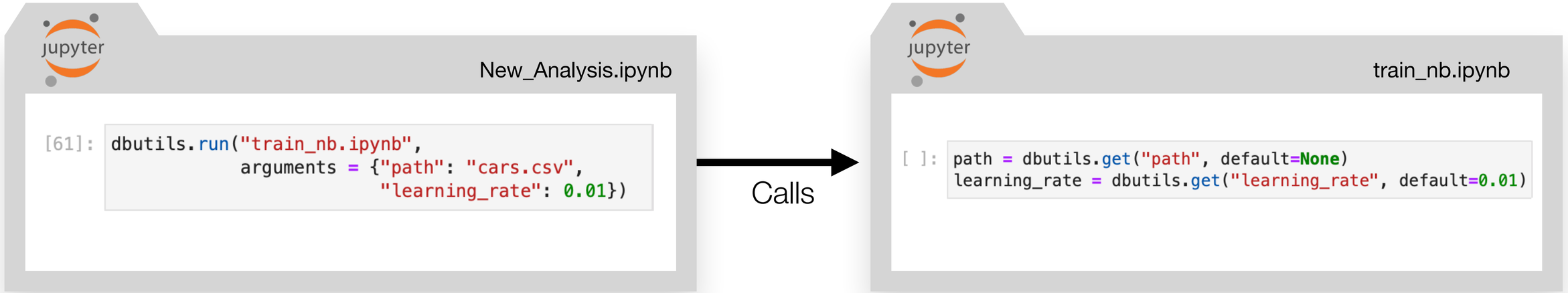
“When is my work good enough to share with other people?” (IP17)

4 **Team** Shared Template Notebooks

Cleaned, generalized notebooks for a task that have been shared to a central store

Template notebooks with TODOs at the top

Using notebooks as functions and libraries



4 **Team** Shared Template Notebooks

Example of data scientists adapting experiment based tool (computational notebook) to support more traditional **function calls and reuse**

Reuse can happen on the platform where work is done such as notebooks

Not all tools easily support this form of reuse so limited adoption (< 40% of survey respondents)

4 **Team** Shared Template Notebooks

Meets people where they work!

Supports easily tweaking existing code:

SP131 uses template notebooks “when the code requires **customization** when applied to different scenarios.”

Drawback: Hard to maintain updates across so many notebook forks / tweaks

5 **Team** Shared Libraries

For code that does not change much (data access APIs)

"The library that we have is for interacting with our catalogs and our file systems. This [notebook] right here is mostly just time series analytics. This could have probably been added to [the library], **but this is something that we add to every day.** Whereas we only change the file system interaction stuff once a month when we need to add a new dataset or something." - IP6

5 **Team** Shared Libraries

Traditional benefits of reusing common code

Some people still prefer the easy **customization** (and debugging) afforded by template notebooks

The nature of data science work does not always lends itself to traditional libraries for analysis code

Influences on Reuse and Sharing

Time savings & time investment

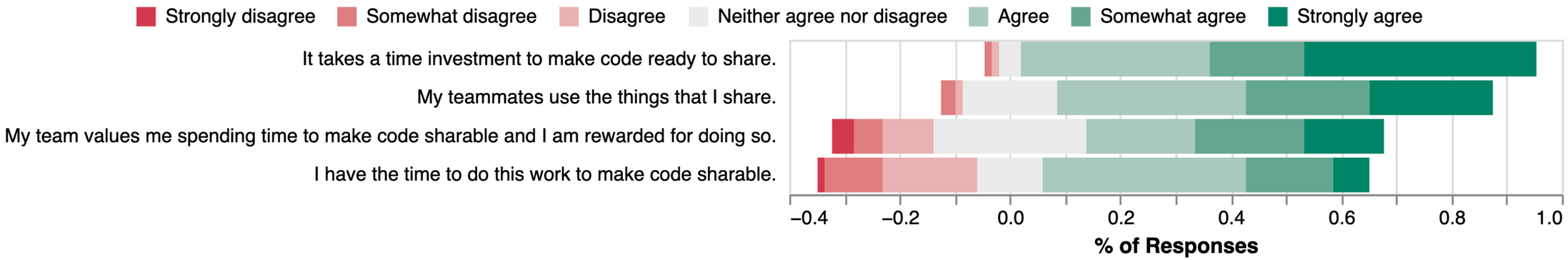
Why do data scientists reuse? **Time savings.**

Why do data scientists *not* reuse? **Time investment.**

Reuse is a more likely when…

- ❖ Cleaner, well-documented code
- ❖ Small number of edits required to adapt
- ❖ Code that would be hard to write on their own

Team culture must incentivize this time investment

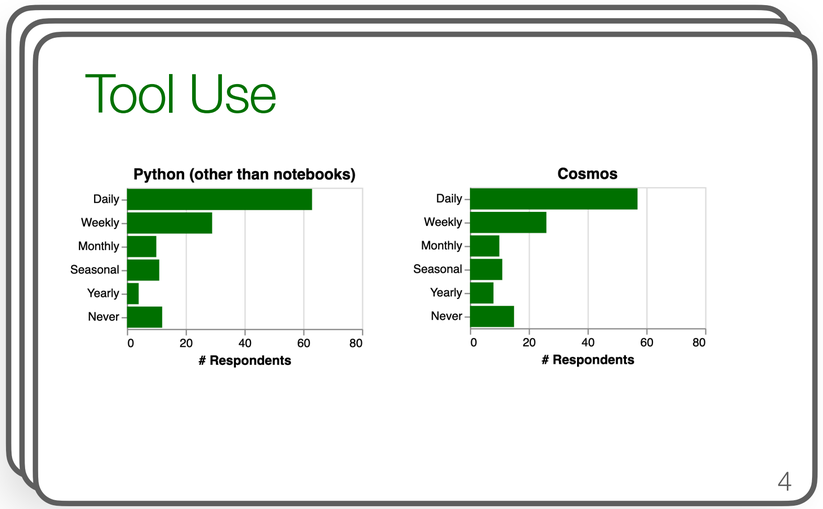


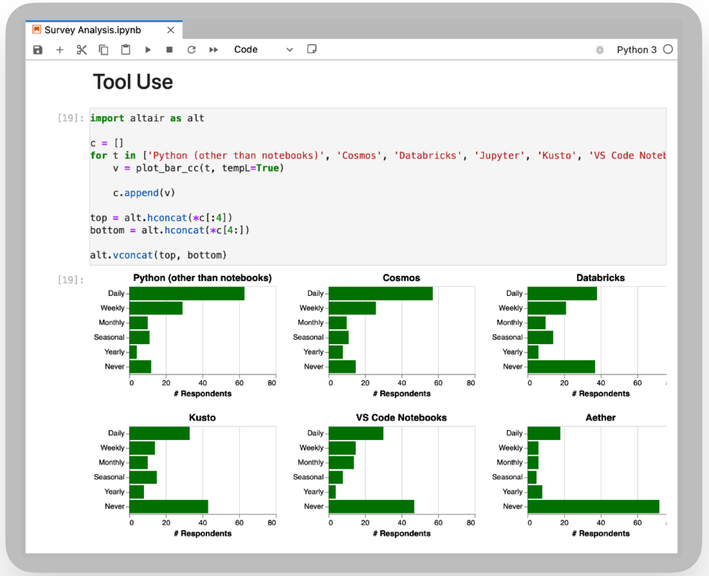
"Sharing code in data science is often not rewarded at the level needed to offset the **investment necessary to do it well.**" - SP104

Implications

Deliverable Is Not Code

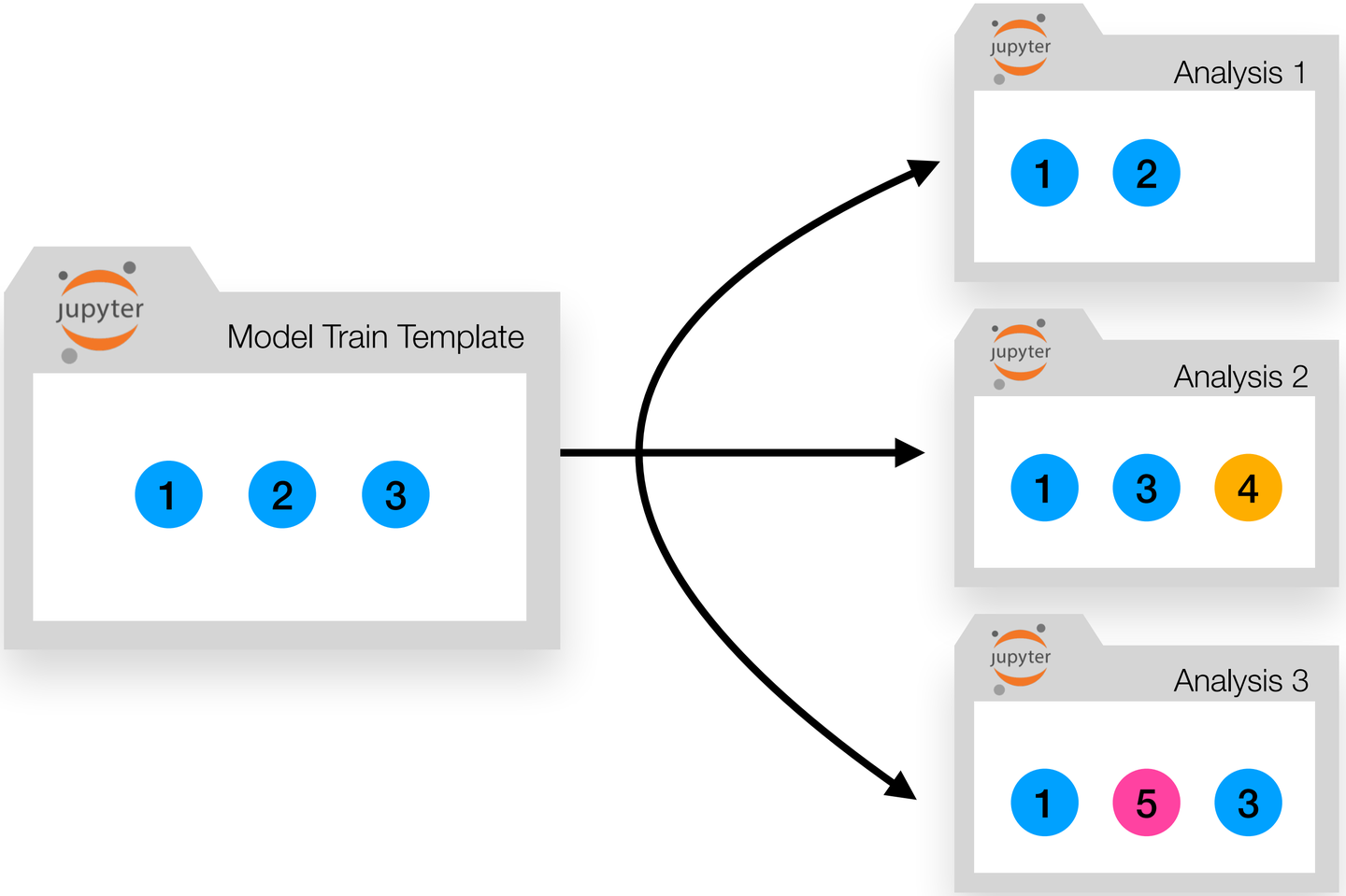
Tighter coupling between presentation environment and coding environment





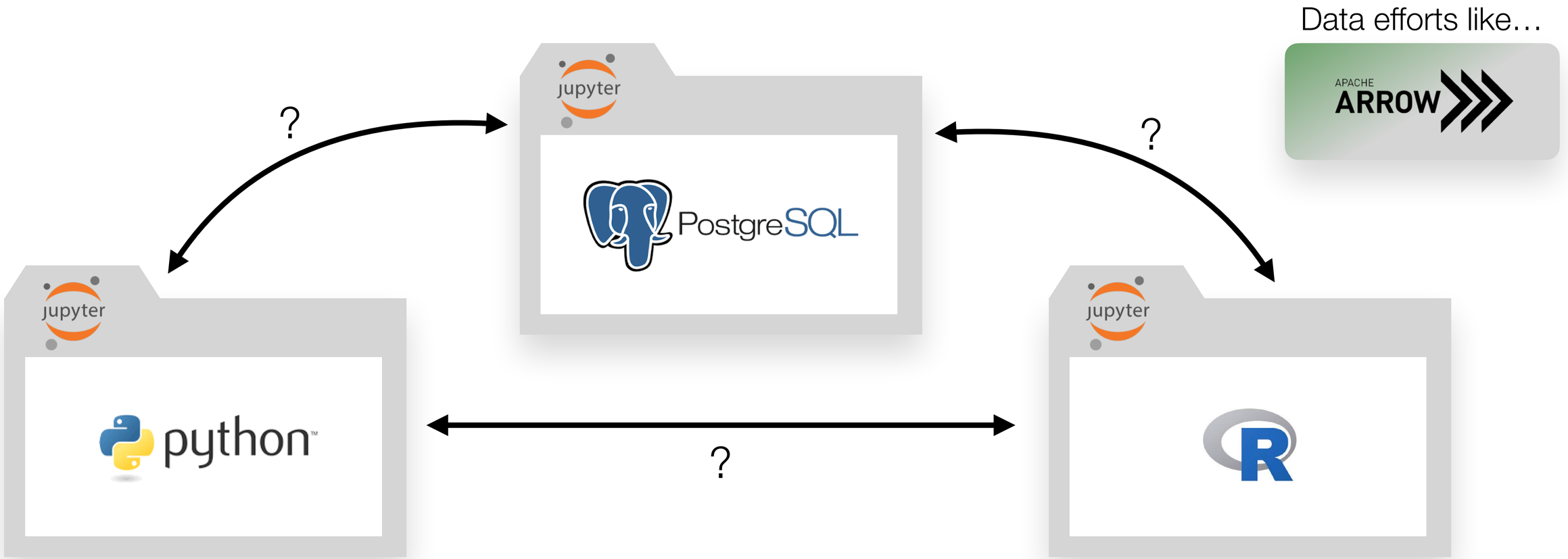
Customization is essential for reuse

Data science code can rarely be reused out of the box, needs to be tweaked



Lack of tool interoperability limits reuse

How can I reuse my Python cleaning script in R or SQL? Or in another tool?



Strategies for Reuse and Sharing among Data Scientists in Software Teams

ICSE SEIP ‘22

Will Epperson
Carnegie Mellon University

April Yi Wang
The University of Michigan

Robert DeLine
Microsoft Research

Steven Drucker
Microsoft Research

Full paper & more!

willepperson.com/papers/reuse-ds

willepp@cmu.edu