Many researchers use code, how many share?

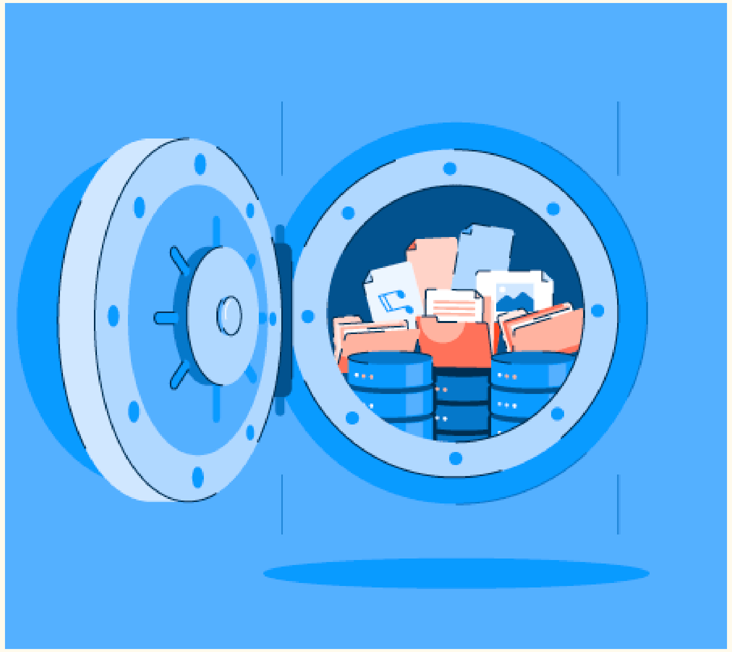
TADA! is a wizard way to enable you to improve code sharing

Dr Joel Pick
University of Edinburgh
&
Dr Ed Ivimey-Cook
University of East Anglia

SORTEE
Society for Open, Reliable, and Transparent
Ecology and Evolutionary Biology

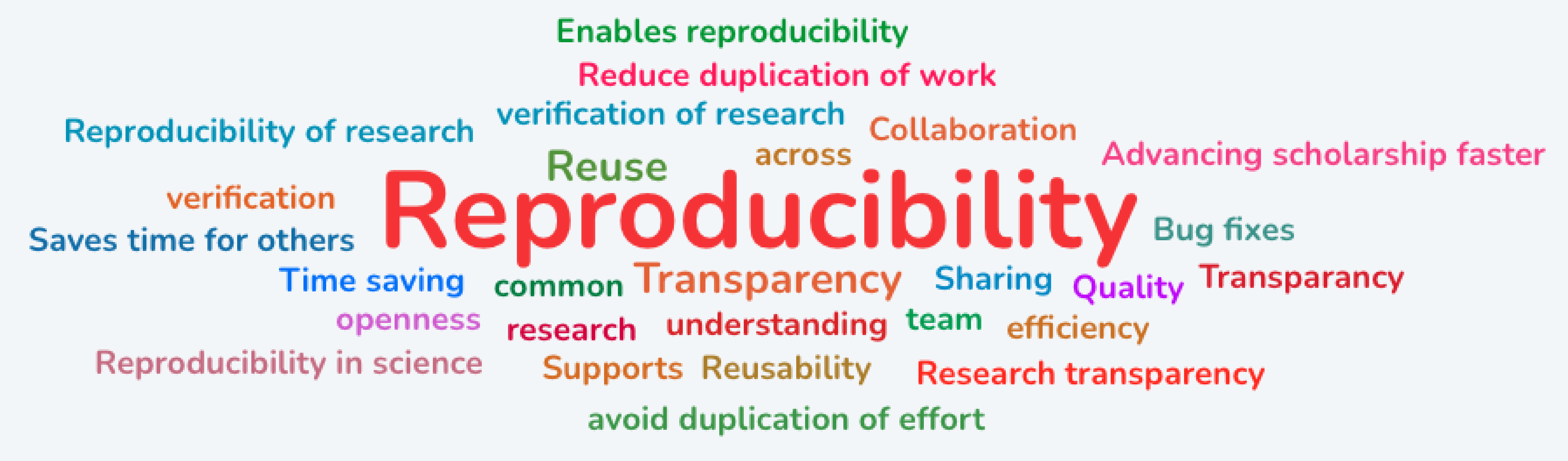
2 – 6 MARCH
OPEN RESEARCH WEEK 2026
Edge Hill University
LIVERPOOL JOHN MOORES UNIVERSITY
UNIVERSITY OF LIVERPOOL
University of Essex

Data and Code Archiving in Ecology and Evolution



- Data archiving high in
  - 35-80% (Culina et al 2020, Kimmel et al 2023, Sánchez-Tójar et al 2025)
  - Clear increase over time
- But quality of archived data low
  - 56.4% complete, 45.9% reusable (Roche et al 2015, 2022)
  - Little increase over time
- Code archiving and code quality remains low
  - Low code availability: 5-30% (Culina et al 2020, Kimmel et al 2023, Sánchez-Tójar et al 2025)
  - Low code functionality: 25% (Trisovic et al 2022, Kellner et al 2025)
  - Low code reproducibility: ~25% (Kambouris et al 2025)

Why is code sharing important?


Enables reproducibility
Reduce duplication of work
Reproducibility of research
verification of research
Collaboration
Advancing scholarship faster
Reuse
across
verification
Reproducibility
Saves time for others
Bug fixes
Time saving
common
Transparency
Sharing
Quality
Transparancy
openness
research
understanding
team
efficiency
Reproducibility in science
Supports
Reusability
Research transparency
avoid duplication of effort

Why is code sharing important?

- **Transparency** of methods
  - See explicitly how the results were generated
  - Able to build on the methods and adapt code for new projects
  - Allows mistakes to be found

- **Computational reproducibility**
  - the ability to obtain the same results given the same data and code

Major way that researchers can earn trust

- We can’t re-examine the data collection process
  - We need to trust authors

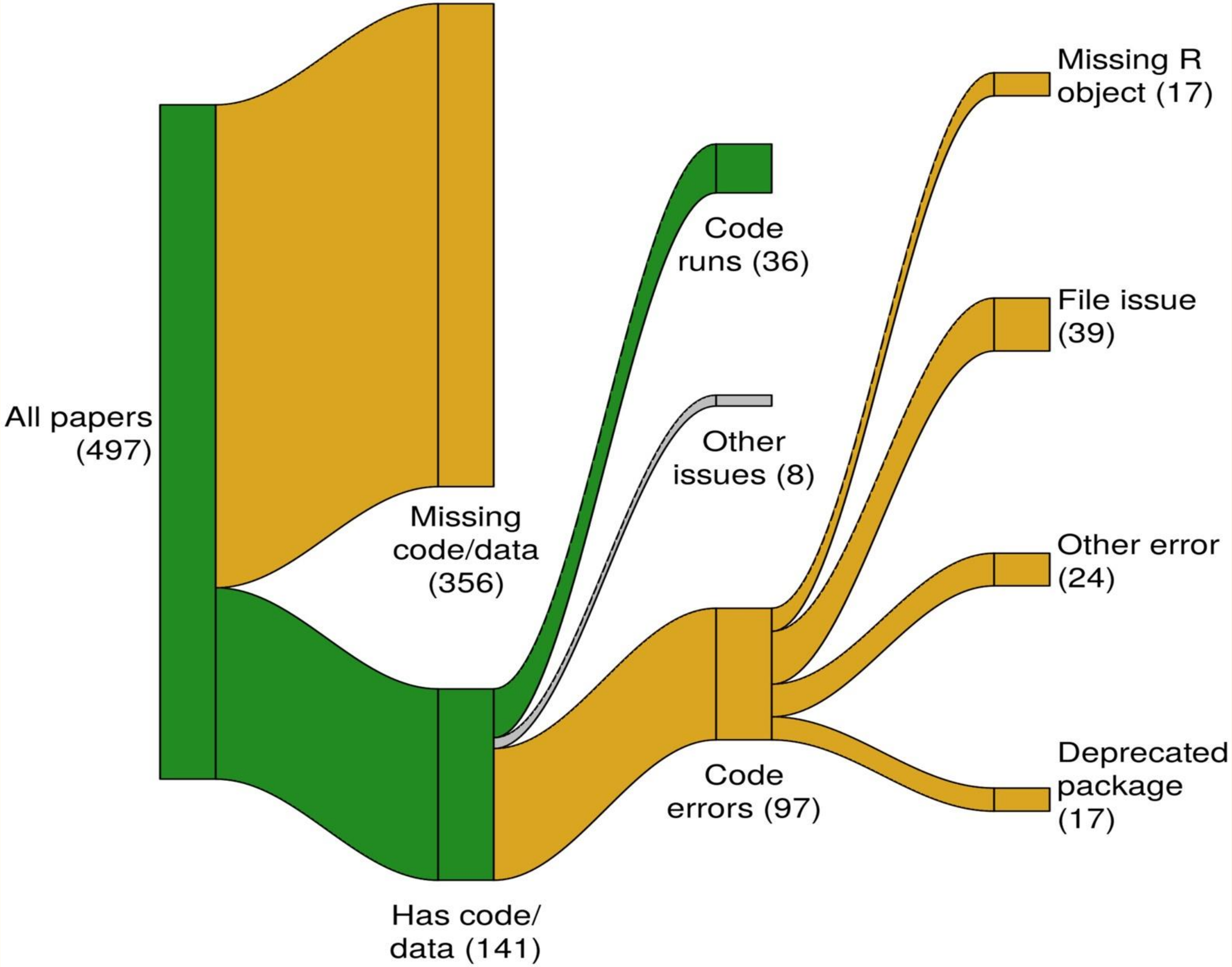
- We can re-examine analytical pathway from data to published results
  - We don’t need to trust
  - Transparent research pipeline

Code sharing is low

Table 1. Studies assessing the code availability in published EcoEvo articles.

Article	Proportion	Time	Studies	Sample
Sánchez-Tójar et al 2025	4.8%	2015-2019	314	12 ecological journals without code-sharing policies
Maitter et al 2024	5.5%	2010-2022	1001	Agriculture and biological science journals
Kambouris et al 2024	16%	2015-2017	177	Meta-analyses in EcoEvo
Kimmel et al 2023	27%	2018-2022	354	5 ecology journals
Culina et al 2020	27%	2015-2019	346	14 ecological journals with code-sharing policies
Kellner et al 2025	33%	2018-2022	497	9 ecology-related journals

When shared, code functionality is often poor



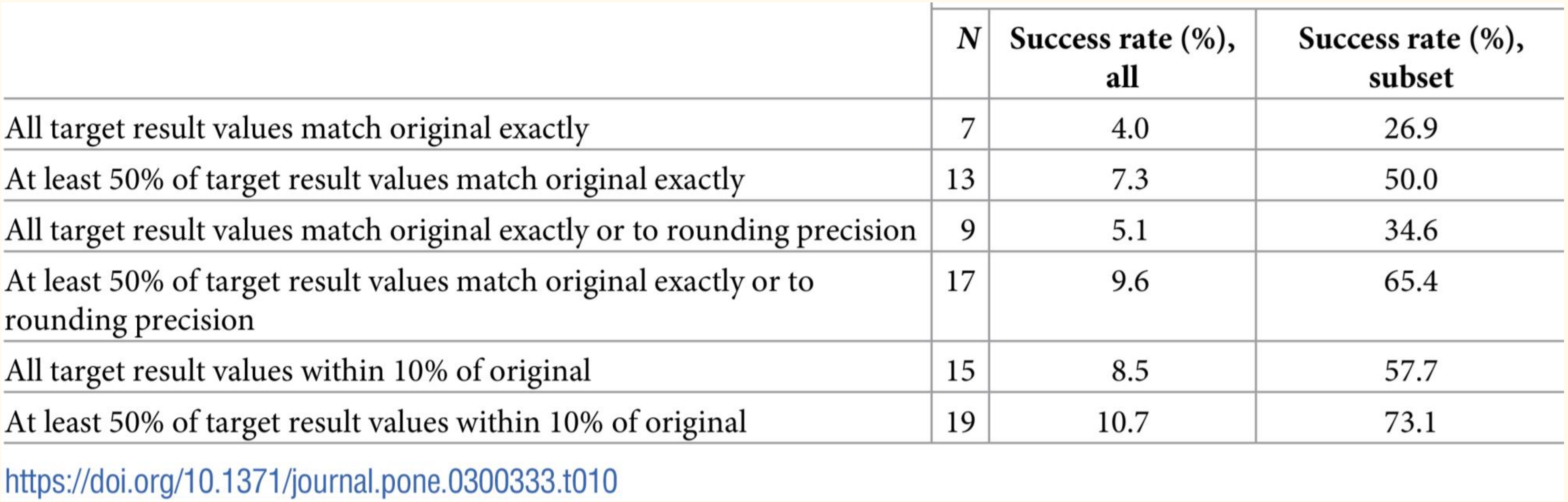
Functionality = code runs with out error.

Kellner et al 2025

Even when shared and functional, code sharing does not ensure computational reproducibility.

From all 177 papers

From 26 papers with data and code



	N	**Success rate (%), all**	**Success rate (%), subset**
All target result values match original exactly	7	4.0	26.9
At least 50% of target result values match original exactly	13	7.3	50.0
All target result values match original exactly or to rounding precision	9	5.1	34.6
At least 50% of target result values match original exactly or to rounding precision	17	9.6	65.4
All target result values within 10% of original	15	8.5	57.7
At least 50% of target result values within 10% of original	19	10.7	73.1

https://doi.org/10.1371/journal.pone.0300333.t010

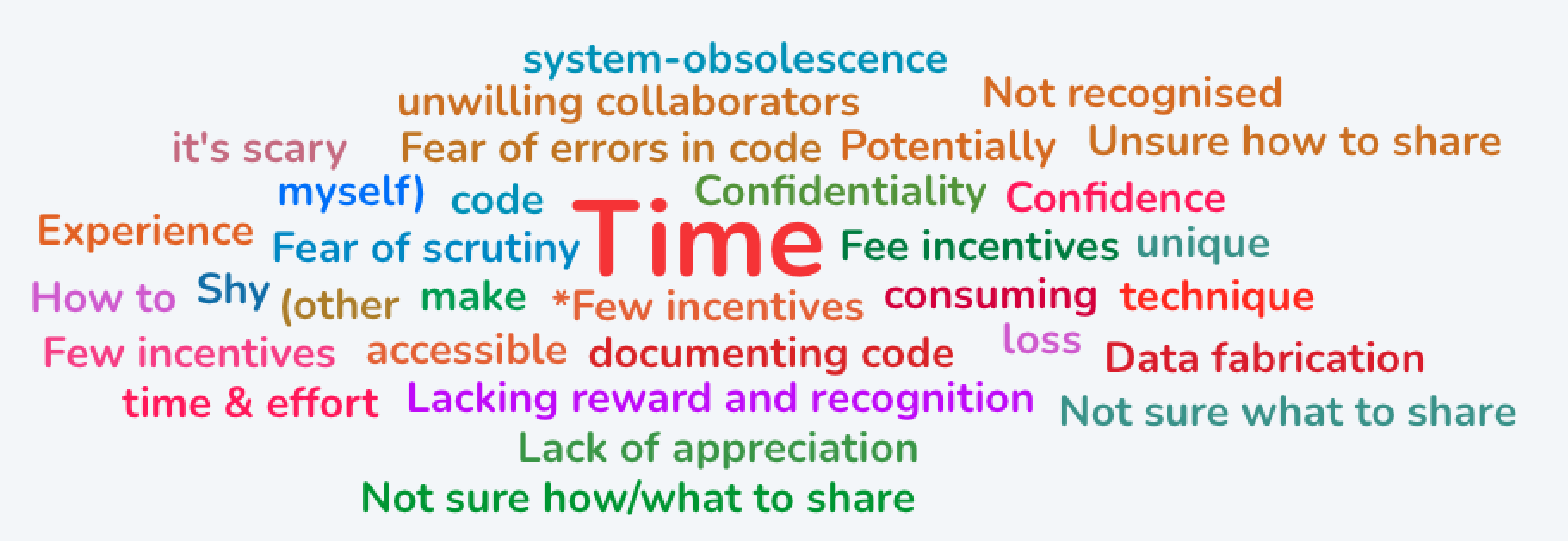
Computational Reproducibility: the ability to obtain the same results given the same data and code

Kambouris et al 2024

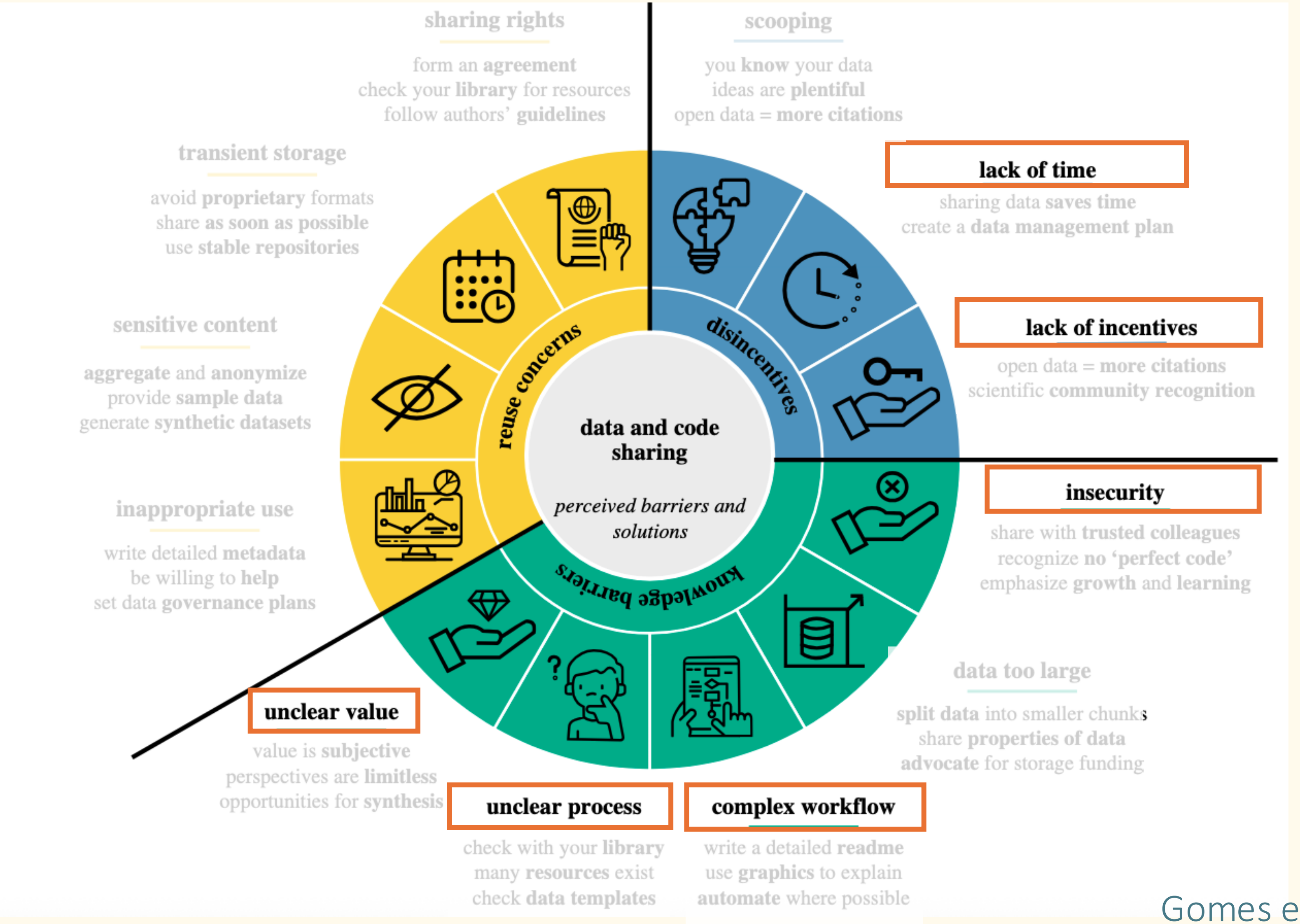
Key Points

- Code is often not shared
- When shared, code has limited functionality
- When shared and functional, results not always reproduceable

What are the barriers to code sharing?



What are the barriers to code sharing?



sharing rights
form an **agreement**
check your **library** for resources
follow authors' **guidelines**

scooping
you **know** your data
ideas are **plentiful**
open data = **more citations**

transient storage
avoid **proprietary** formats
share **as soon as possible**
use **stable repositories**

lack of time
sharing data **saves time**
create a **data management plan**

sensitive content
aggregate and **anonymize**
provide **sample data**
generate **synthetic datasets**

lack of incentives
open data = **more citations**
scientific **community recognition**

inappropriate use
write detailed **metadata**
be willing to **help**
set data **governance plans**

insecurity
share with **trusted colleagues**
recognize **no 'perfect code'**
emphasize **growth** and **learning**

data too large
split data into smaller chunks
share properties of data
advocate for storage funding

unclear value
value is **subjective**
perspectives are **limitless**
opportunities for **synthesis**

unclear process
check with your **library**
many **resources** exist
check **data templates**

complex workflow
write a detailed **readme**
use **graphics** to explain
automate where possible

Gomes et al. 2022 *Proc Roy Soc B*

How can we increase code sharing?

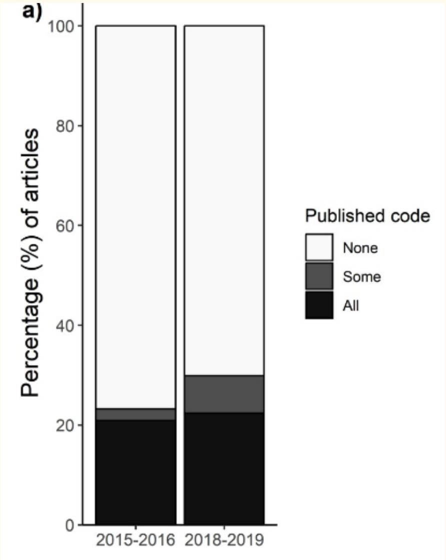
How can we increase code sharing?

- Change policy at Funders and Journals
- Education and improving practice

Clear journal guidelines and mandatory code archiving

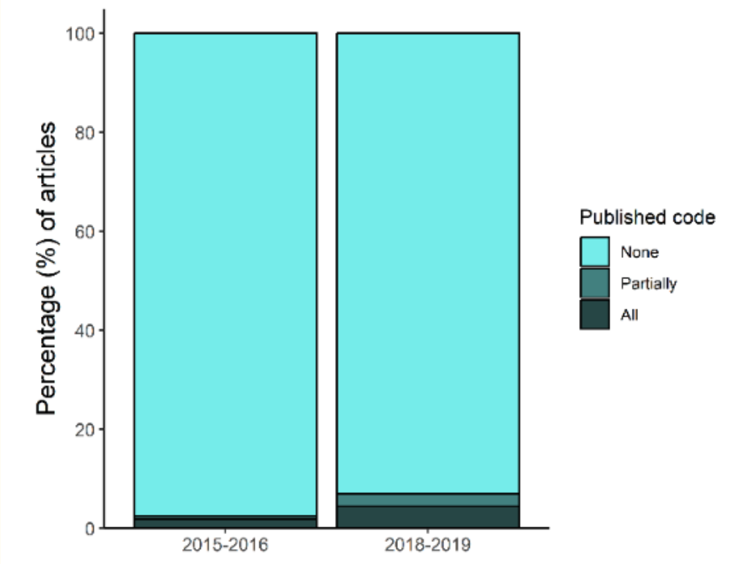
- Code sharing policy associated with higher code availability

Journals with code policy



Culina et al 2020 *PLoS Biology*

Journals without code policy



Sánchez-Tójar et al 2025 *Peer Community Journal*

- Many more journals could do this
  - 26.6% encouraged and 26.9% mandated code-sharing

Ivimey-Cook et al. 2025 *Proc. Roy. Soc. B*

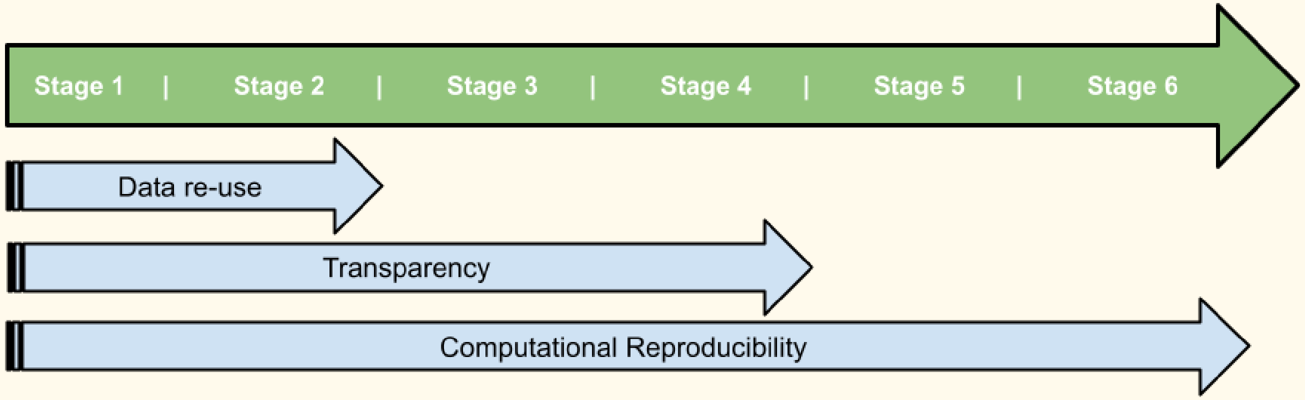
Data and Code Editors at Journals

- Assess the quality of archived data and code
- Help authors comply to journal mandates
- Standardized guidelines

Peer Community Journal

Section: Ecology

The SORTEE guidelines for data and code quality control in ecology and evolutionary biology



The American Naturalist

Ecology Letters

Behavioral Ecology

Journal of Evolutionary Biology

Proceedings of the Royal Society B
Biological Sciences

PCI
Peer Community in

[Pick et al. 2026 *Peer Community Journal*]

Education and improving practice

- Low sharing maybe linked to lack of knowledge of how to share functional and transparent code.

- Current guidelines are often difficult to follow, geared towards software developers and not analytical research code, or do not follow FAIR principles

F A I R

Findable

Accessible

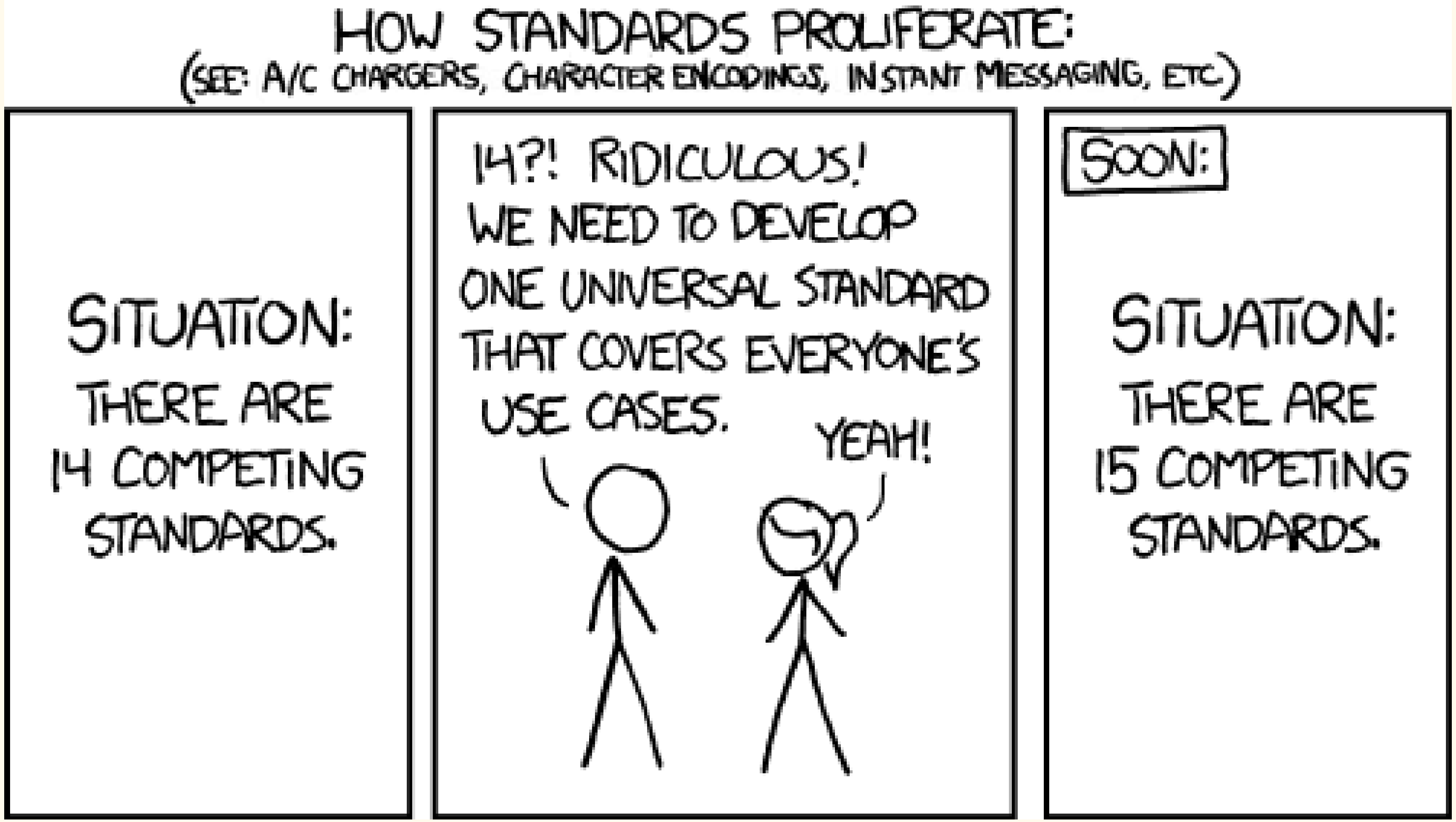
Interoperable

Reusable

Introducing TADA!

Transferable
Available
Documented
Annotated

Ivimey-Cook et al. 2025 *EcoEvoRxiv*

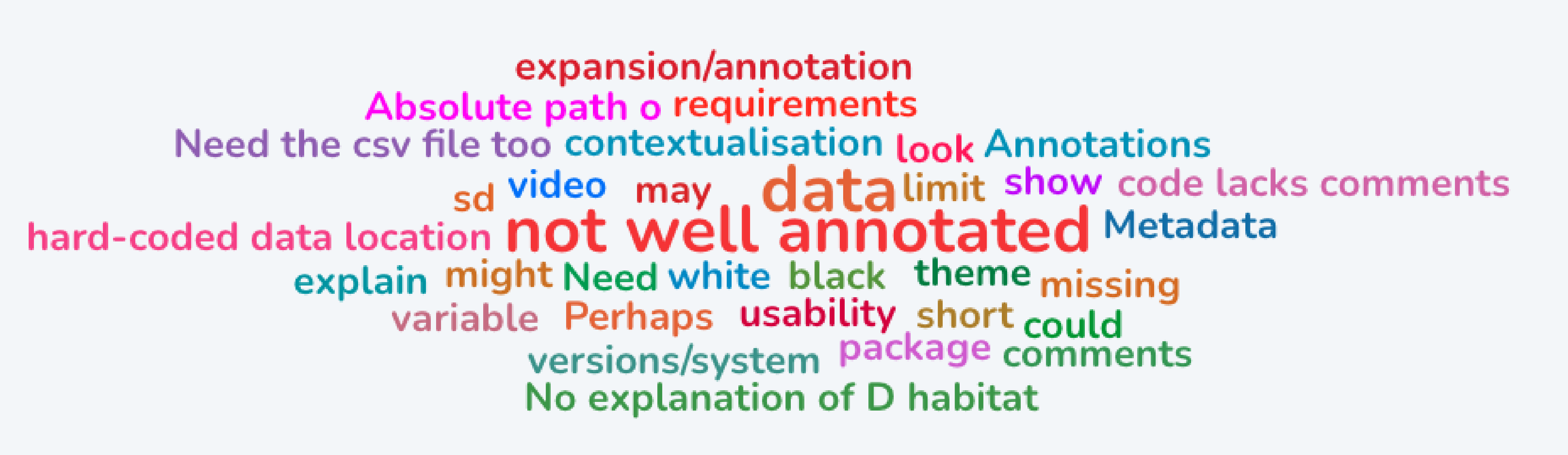


https://xkcd.com/927/

Have a look at the code in MyCode.pdf

1. What are some immediate things you notice?

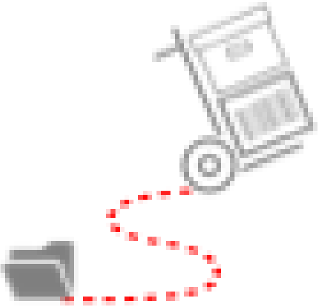
2. What could we do to improve how it is shared?



Transferable

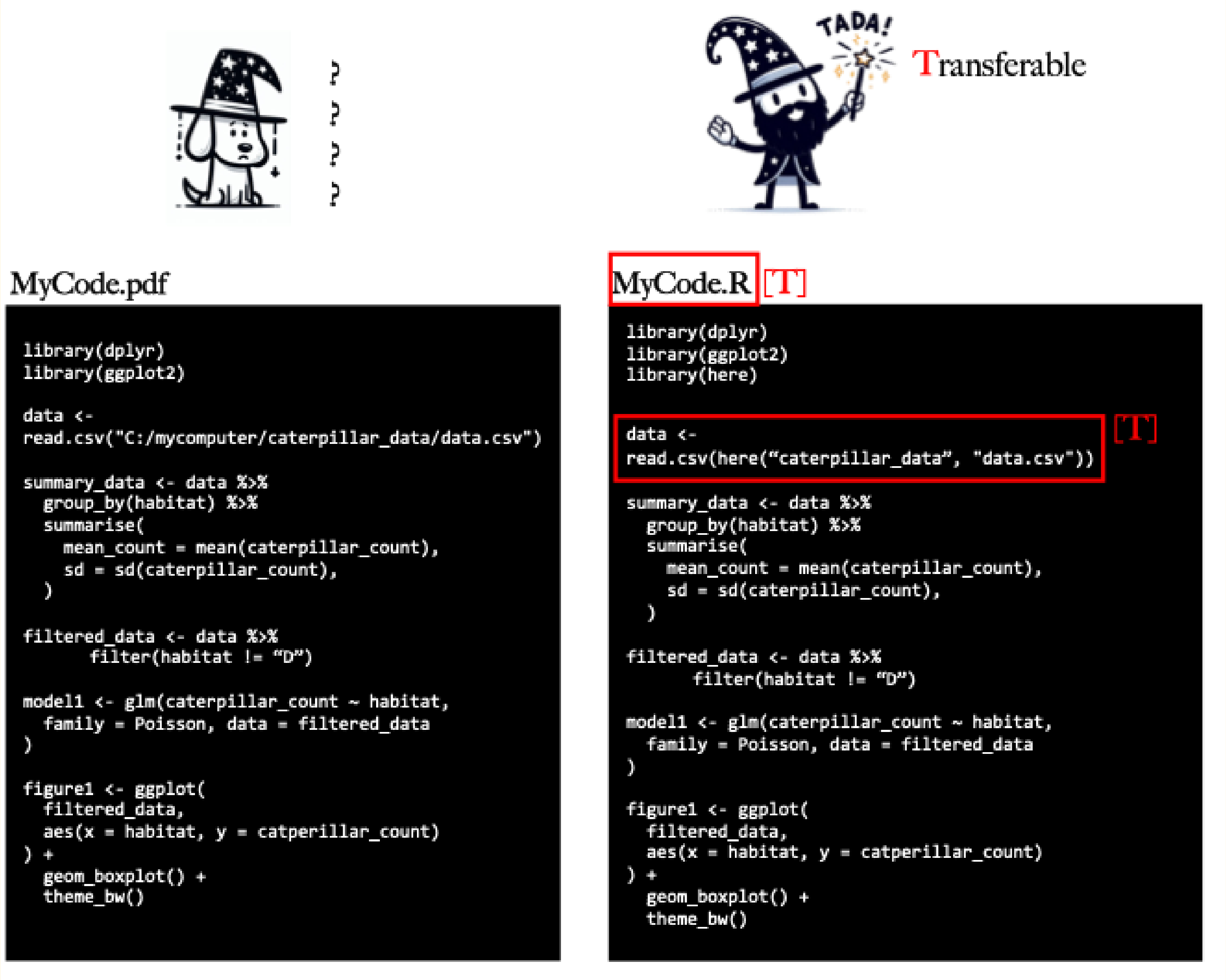
Anyone can open the file, view the code, and run the script without needing to convert the file or alter the code!

How could we make this code more “Transferable” between users?



- ☐ *file formats* → *Use interoperable file formats (e.g. .R, .py, .cpp)*
- ☐ *file paths* → *Use relative file paths and tools that can prevent coding user-specific paths*

Transferable



Transferable

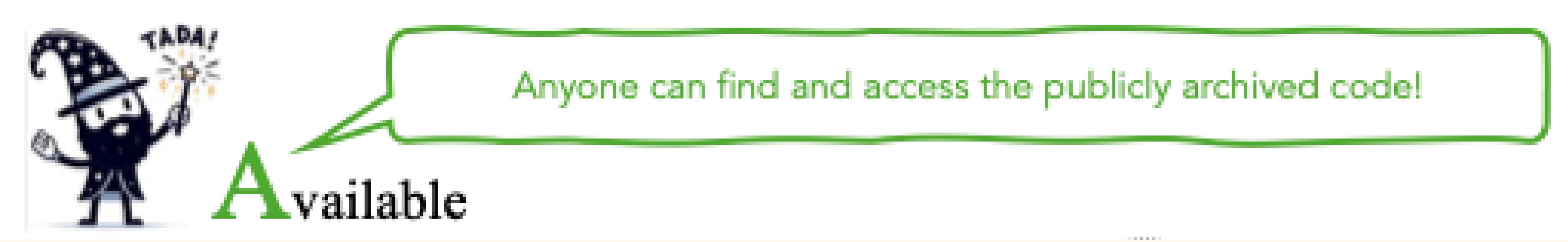
MyCode.pdf

```
library(dplyr)
library(ggplot2)

data <-
read.csv("C:/mycomputer/caterpillar_data/data.csv")

summary_data <- data %>%
  group_by(habitat) %>%
  summarise(
    mean_count = mean(caterpillar_count),
    sd = sd(caterpillar_count),
  )

filtered_data <- data %>%
       filter(habitat != “D”)

model1 <- glm(caterpillar_count ~ habitat,
  family = Poisson, data = filtered_data
)

figure1 <- ggplot(
  filtered_data,
  aes(x = habitat, y = catperillar_count)
) +
  geom_boxplot() +
  theme_bw()
```

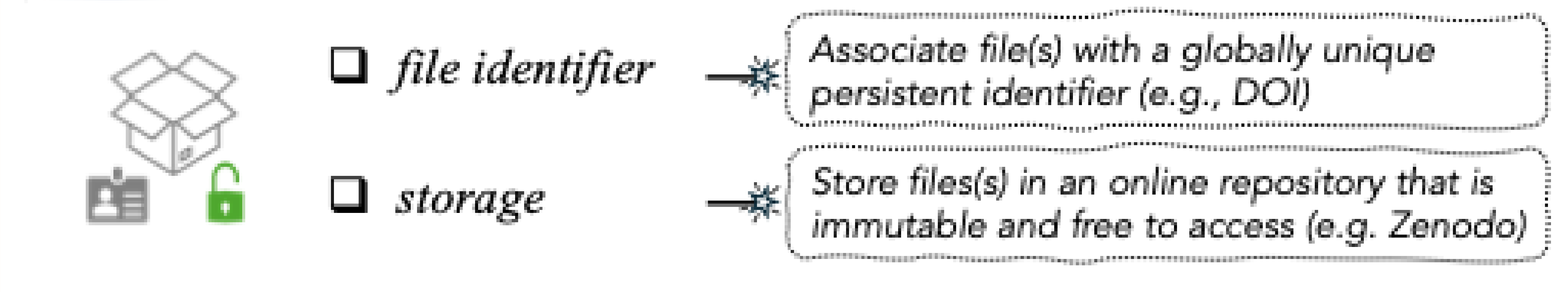
MyCode.R [T]

```
library(dplyr)
library(ggplot2)
library(here)

data <-
read.csv(here(“caterpillar_data”, "data.csv"))

summary_data <- data %>%
  group_by(habitat) %>%
  summarise(
    mean_count = mean(caterpillar_count),
    sd = sd(caterpillar_count),
  )

filtered_data <- data %>%
       filter(habitat != “D”)

model1 <- glm(caterpillar_count ~ habitat,
  family = Poisson, data = filtered_data
)

figure1 <- ggplot(
  filtered_data,
  aes(x = habitat, y = catperillar_count)
) +
  geom_boxplot() +
  theme_bw()
```

[T]



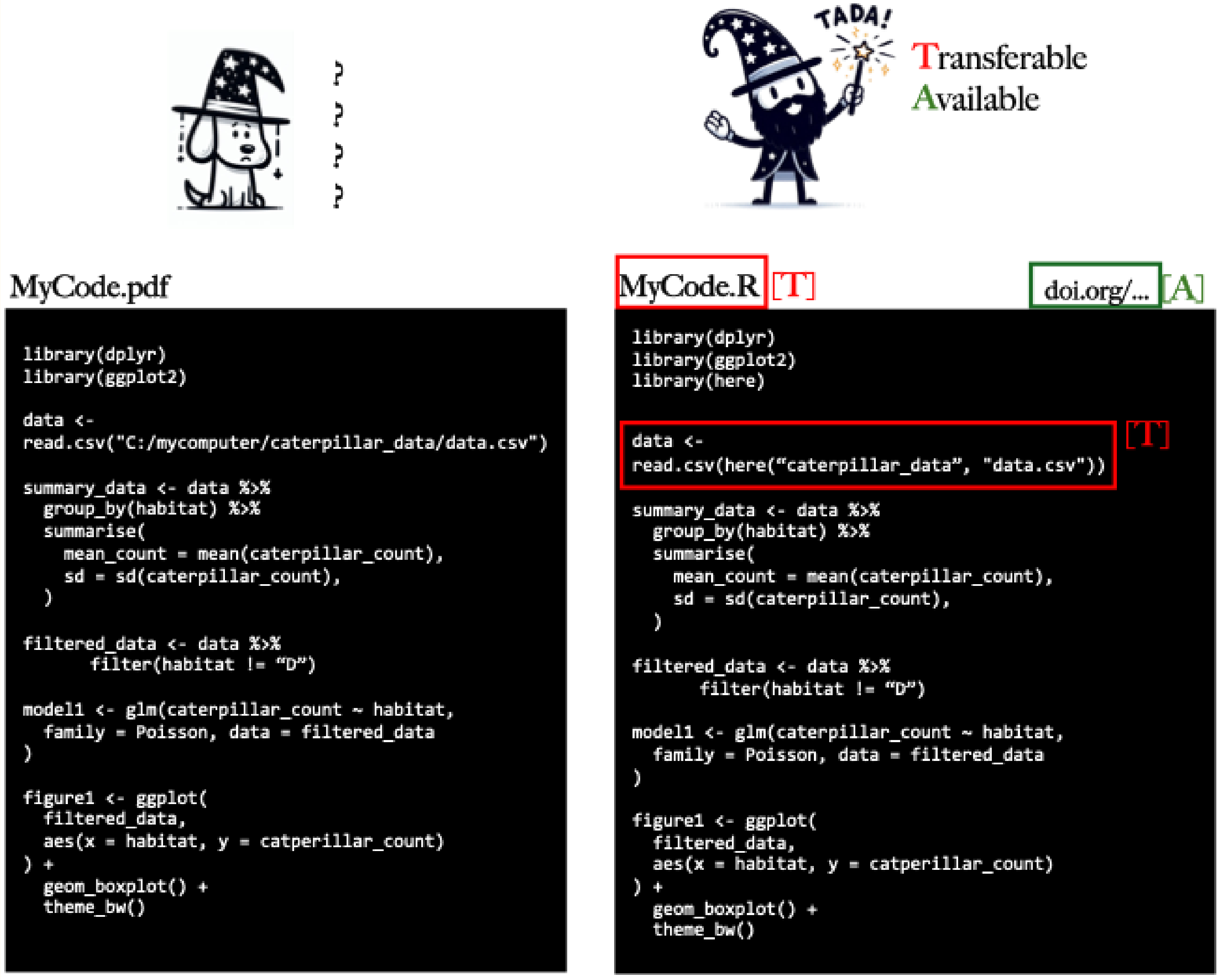
How could we make this code “Available”?



☐ *file identifier* → Associate file(s) with a globally unique persistent identifier (e.g., DOI)

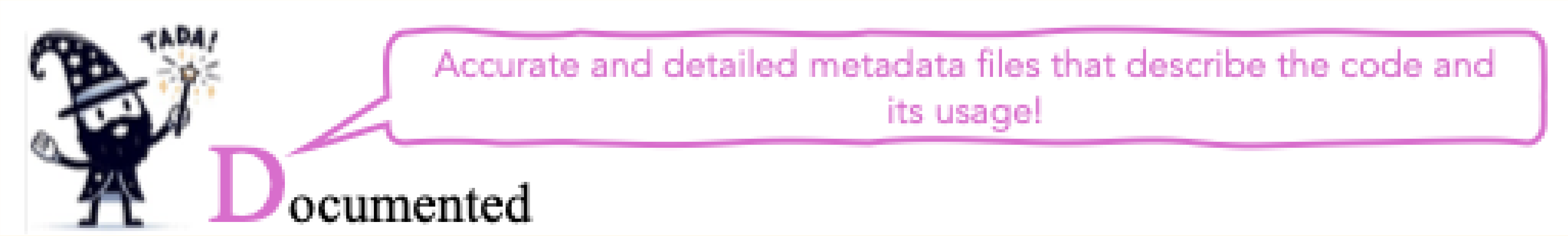
☐ *storage* → Store files(s) in an online repository that is immutable and free to access (e.g. Zenodo)

Available



```
library(dplyr)
library(ggplot2)

data <-
read.csv("C:/mycomputer/caterpillar_data/data.csv")

summary_data <- data %>%
  group_by(habitat) %>%
  summarise(
    mean_count = mean(caterpillar_count),
    sd = sd(caterpillar_count),
  )

filtered_data <- data %>%
      filter(habitat != “D”)

model1 <- glm(caterpillar_count ~ habitat,
  family = Poisson, data = filtered_data
)

figure1 <- ggplot(
  filtered_data,
  aes(x = habitat, y = catperillar_count)
) +
  geom_boxplot() +
  theme_bw()
```

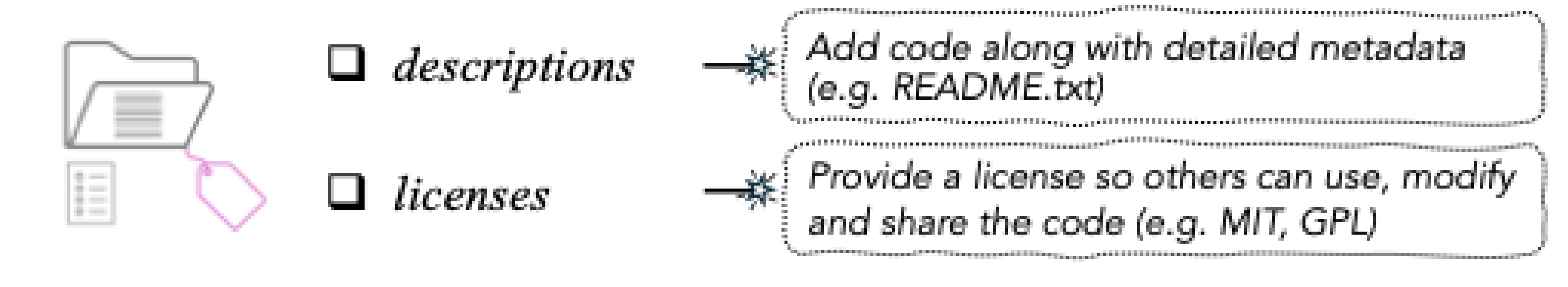
```
library(dplyr)
library(ggplot2)
library(here)

data <-
read.csv(here(“caterpillar_data”, "data.csv"))

summary_data <- data %>%
  group_by(habitat) %>%
  summarise(
    mean_count = mean(caterpillar_count),
    sd = sd(caterpillar_count),
  )

filtered_data <- data %>%
      filter(habitat != “D”)

model1 <- glm(caterpillar_count ~ habitat,
  family = Poisson, data = filtered_data
)

figure1 <- ggplot(
  filtered_data,
  aes(x = habitat, y = catperillar_count)
) +
  geom_boxplot() +
  theme_bw()
```



Documented

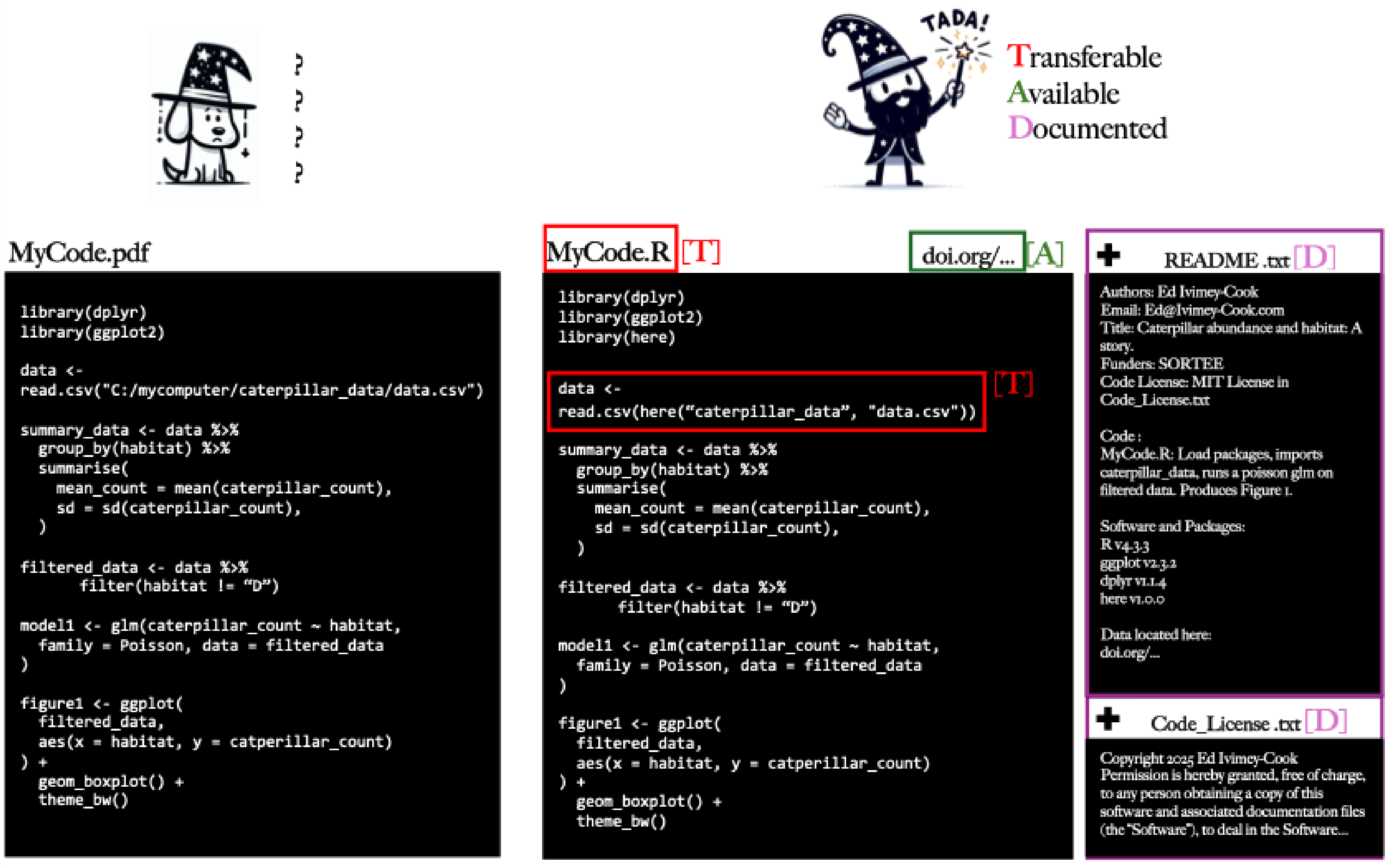
Accurate and detailed metadata files that describe the code and its usage!

How could we “document” this code?



- ☐ *descriptions* → *Add code along with detailed metadata (e.g. README.txt)*
- ☐ *licenses* → *Provide a license so others can use, modify and share the code (e.g. MIT, GPL)*

Documented



Transferable
Available
Documented

MyCode.pdf

```
library(dplyr)
library(ggplot2)

data <-
read.csv("C:/mycomputer/caterpillar_data/data.csv")

summary_data <- data %>%
  group_by(habitat) %>%
  summarise(
    mean_count = mean(caterpillar_count),
    sd = sd(caterpillar_count),
  )

filtered_data <- data %>%
       filter(habitat != “D”)

model1 <- glm(caterpillar_count ~ habitat,
  family = Poisson, data = filtered_data
)

figure1 <- ggplot(
  filtered_data,
  aes(x = habitat, y = catperillar_count)
) +
  geom_boxplot() +
  theme_bw()
```

MyCode.R [T] doi.org/... [A]

```
library(dplyr)
library(ggplot2)
library(here)

data <-
read.csv(here(“caterpillar_data”, "data.csv"))

summary_data <- data %>%
  group_by(habitat) %>%
  summarise(
    mean_count = mean(caterpillar_count),
    sd = sd(caterpillar_count),
  )

filtered_data <- data %>%
       filter(habitat != “D”)

model1 <- glm(caterpillar_count ~ habitat,
  family = Poisson, data = filtered_data
)

figure1 <- ggplot(
  filtered_data,
  aes(x = habitat, y = catperillar_count)
) +
  geom_boxplot() +
  theme_bw()
```

[T]

✚ README .txt [D]

Authors: Ed Ivimey-Cook
Email: Ed@Ivimey-Cook.com
Title: Caterpillar abundance and habitat: A story.
Funders: SORTEE
Code License: MIT License in Code_License.txt

Code :
MyCode.R: Load packages, imports caterpillar_data, runs a poisson glm on filtered data. Produces Figure 1.

Software and Packages:
R v4.3.3
ggplot v2.3.2
dplyr v1.1.4
here v1.0.0

Data located here:
doi.org/...

✚ Code_License .txt [D]

Copyright 2025 Ed Ivimey-Cook
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software...

Comments within the code that explain what it does, why, how and what it produces!

Annotated

How could we make this code “Annotated”?

- ☐ *code comments* → Divide code into logical sections (chunks) and describe these sections
- ☐ *markdown files* → Embed code and text in markdown files (e.g. Quarto, Jupyter Notebook)

Annotated

Transferable
Available
Documented
Annotated

MyCode.pdf

```
library(dplyr)
library(ggplot2)

data <-
read.csv("C:/mycomputer/caterpillar_data/data.csv")

summary_data <- data %>%
  group_by(habitat) %>%
  summarise(
    mean_count = mean(caterpillar_count),
    sd = sd(caterpillar_count),
  )

filtered_data <- data %>%
       filter(habitat != “D”)

model1 <- glm(caterpillar_count ~ habitat,
  family = Poisson, data = filtered_data
)

figure1 <- ggplot(
  filtered_data,
  aes(x = habitat, y = catperillar_count)
) +
  geom_boxplot() +
  theme_bw()
```

MyCode.R [T]

doi.org/... [A]

```
# Load packages#####
library(dplyr)
library(ggplot2)
library(here)

# Load caterpillar abundance data (w/o local file paths)#####
data <- read.csv(here("caterpillar_data", "data.csv"))

# summarise the mean number caterpillars with error#####
summary_data <- data %>%
  group_by(habitat) %>%
  summarise(
    mean_count = mean(caterpillar_count),
    sd = sd(caterpillar_count),
  )

# Remove values from habitat D as these are an error#####
filtered_data <- data %>%
        filter(habitat != “D”)

# Run a Poisson general linear model######
# to analyse caterpillar abundance varying with habitat
# numeric results in “Caterpillar Abundance”
model1 <- glm(caterpillar_count ~ habitat,
  family = Poisson, data = filtered_data
)

#create figure 1, caterpillar count against habitat####
figure1 <- ggplot(
  filtered_data,
  aes(x = habitat, y = caterpillar_count)
) +
  geom_boxplot() +
  theme_bw()
```

[T]

[A]

✚ README .txt [D]

Authors: Ed Ivimey-Cook
Email: Ed@Ivimey-Cook.com
Title: Caterpillar abundance and habitat: A story.
Funders: SORTEE
Code License: MIT License in Code_License.txt

Code :
MyCode.R: Load packages, imports caterpillar_data, runs a poisson glm on filtered data. Produces Figure 1.

Software and Packages:
R v4.3.3
ggplot v2.3.2
dplyr v1.1.4
here v1.0.0

Data located here:
doi.org/...

✚ Code_License .txt [D]

Copyright 2025 Ed Ivimey-Cook
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software...

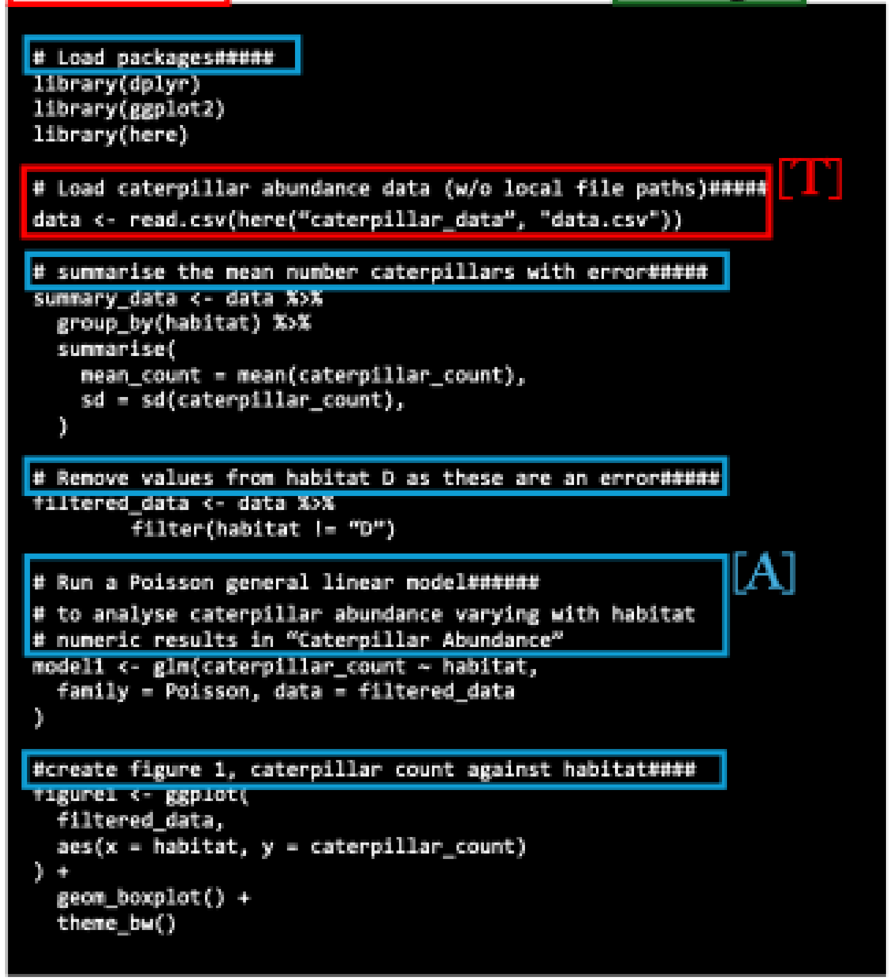
Transferable
Available
Documented
Annotated

MyCode.pdf

```
library(dplyr)
library(ggplot2)

data <-
read.csv("C:/mycomputer/caterpillar_data/data.csv")

summary_data <- data %>%
  group_by(habitat) %>%
  summarise(
    mean_count = mean(caterpillar_count),
    sd = sd(caterpillar_count),
  )

filtered_data <- data %>%
       filter(habitat != "D")

model1 <- glm(caterpillar_count ~ habitat,
  family = Poisson, data = filtered_data
)

figure1 <- ggplot(
  filtered_data,
  aes(x = habitat, y = catperillar_count)
) +
  geom_boxplot() +
  theme_bw()
```

MyCode.R [T]

doi.org/... [A]



```
# Load packages#####
library(dplyr)
library(ggplot2)
library(here)

# Load caterpillar abundance data (w/o local file paths)#####
data <- read.csv(here("caterpillar_data", "data.csv"))

# summarise the mean number caterpillars with error#####
summary_data <- data %>%
  group_by(habitat) %>%
  summarise(
    mean_count = mean(caterpillar_count),
    sd = sd(caterpillar_count),
  )

# Remove values from habitat D as these are an error#####
filtered_data <- data %>%
        filter(habitat != "D")

# Run a Poisson general linear model######
# to analyse caterpillar abundance varying with habitat
# numeric results in "Caterpillar Abundance"
model1 <- glm(caterpillar_count ~ habitat,
  family = Poisson, data = filtered_data
)

#create figure 1, caterpillar count against habitat####
figure1 <- ggplot(
  filtered_data,
  aes(x = habitat, y = caterpillar_count)
) +
  geom_boxplot() +
  theme_bw()
```

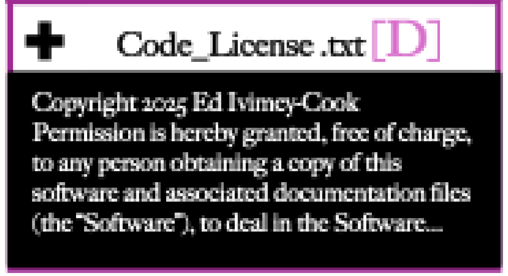
✚ README .txt [D]

Authors: Ed Ivimey-Cook
Email: Ed@Ivimey-Cook.com
Title: Caterpillar abundance and habitat: A story.
Funders: SORTEE
Code License: MIT License in Code_License.txt

Code :
MyCode.R: Load packages, imports caterpillar_data, runs a poisson glm on filtered data. Produces Figure 1.

Software and Packages:
R v4.3.3
ggplot v2.3.2
dplyr v1.1.4
here v1.0.0

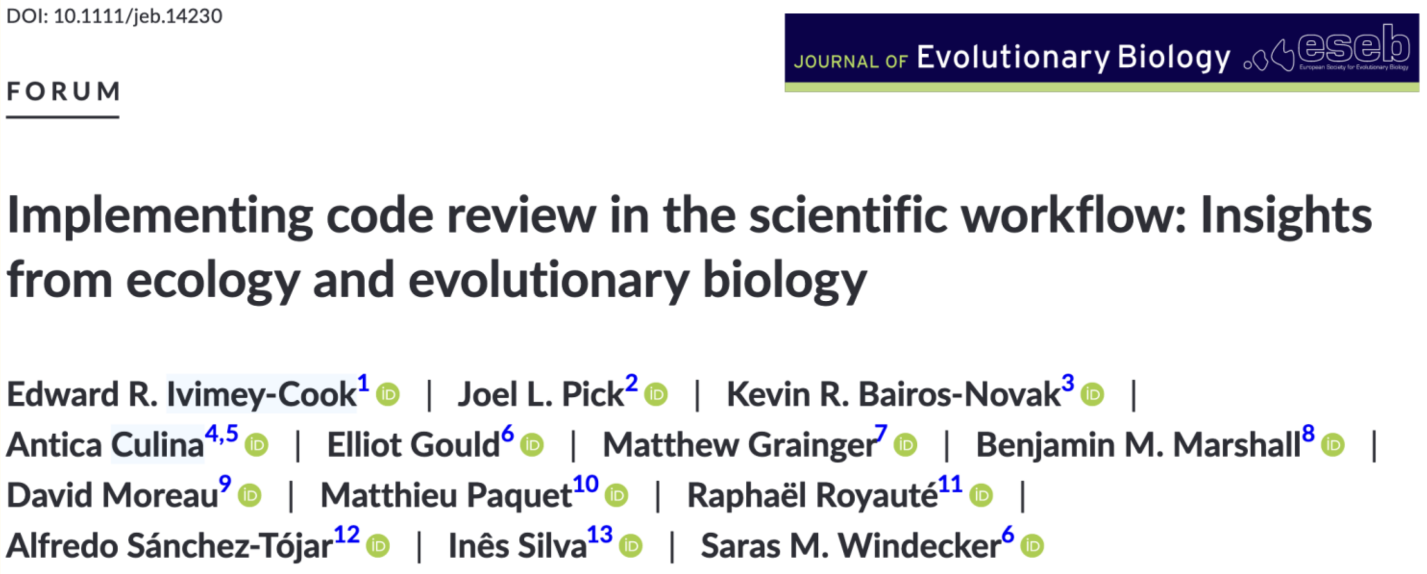
Data located here:
doi.org/...



✚ Code_License .txt [D]

Copyright 2025 Ed Ivimey-Cook
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software...

Code Review

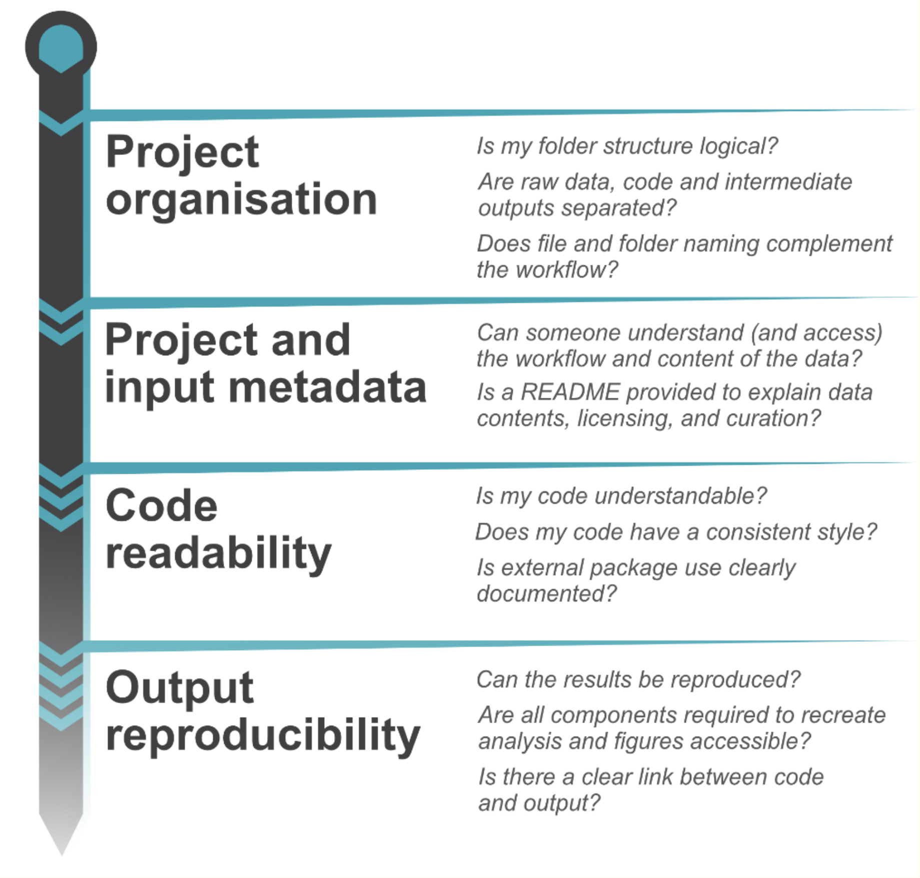

DOI: 10.1111/jeb.14230

JOURNAL OF Evolutionary Biology eseb

FORUM

Implementing code review in the scientific workflow: Insights from ecology and evolutionary biology

Edward R. Ivimey-Cook[1] | Joel L. Pick[2] | Kevin R. Bairos-Novak[3] |
Antica Culina[4,5] | Elliot Gould[6] | Matthew Grainger[7] | Benjamin M. Marshall[8] |
David Moreau[9] | Matthieu Paquet[10] | Raphaël Royauté[11] |
Alfredo Sánchez-Tójar[12] | Inês Silva[13] | Saras M. Windecker[6]



Project organisation

Is my folder structure logical?

Are raw data, code and intermediate outputs separated?

Does file and folder naming complement the workflow?

Project and input metadata

Can someone understand (and access) the workflow and content of the data?

Is a README provided to explain data contents, licensing, and curation?

Code readability

Is my code understandable?

Does my code have a consistent style?

Is external package use clearly documented?

Output reproducibility

Can the results be reproduced?

Are all components required to recreate analysis and figures accessible?

Is there a clear link between code and output?

Ivimey-Cook et al. 2023 *J. Evol. Biol.*

Acknowledgements

- Ed Ivimey-Cook

- Antica Culina, Shreya Dimri , Matthew Grainger, Fonti Kar, Malgorzata Lagisz , Nicholas Patrick Moran , Shinichi Nakagawa , Dominique Roche, Sean Tattan, Alfredo Sanchez-Tojar, Saras Windecker

SORTEE

Society for Open, Reliable, and Transparent Ecology and Evolutionary Biology