# Reproducible research

Karl Broman

Biostatistics & Medical Informatics
University of Wisconsin–Madison

kbroman.org
github.com/kbroman
@kbroman@rstats.me

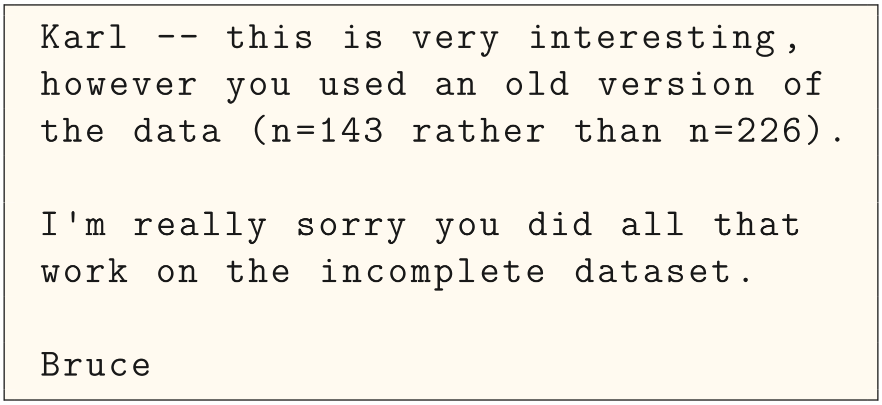
Slides: kbroman.org/Talk_JAXsymp

This is a much-shortened version of lecture of my usual lecture on “steps to reproducible research” (see https://github.com/kbroman/Talk_ReproRes).

This is for a Jackson Lab symposium on 6 Nov 2025, part of the preparations to develop a course on omics analysis.

Source: https://github.com/kbroman/Talk_JAXsymp
These slides: https://kbroman.org/Talk_JAXsymp
Slides with notes: https://kbroman.org/Talk_JAXsymp/jax_symp_withnotes.pdf

By “reproducible research,” I’m referring to “computational reproducibility,” by which I mean that the data and code for a project are packaged together in a way that they can be handed to someone else, who can rerun the code and get the same results—the same figures and tables. This is surprisingly hard to do, and it’s even more difficult in the context of a collaboration between two or more data analysts.



```
Karl -- this is very interesting,
however you used an old version of
the data (n=143 rather than n=226).

I'm really sorry you did all that
work on the incomplete dataset.

Bruce
```

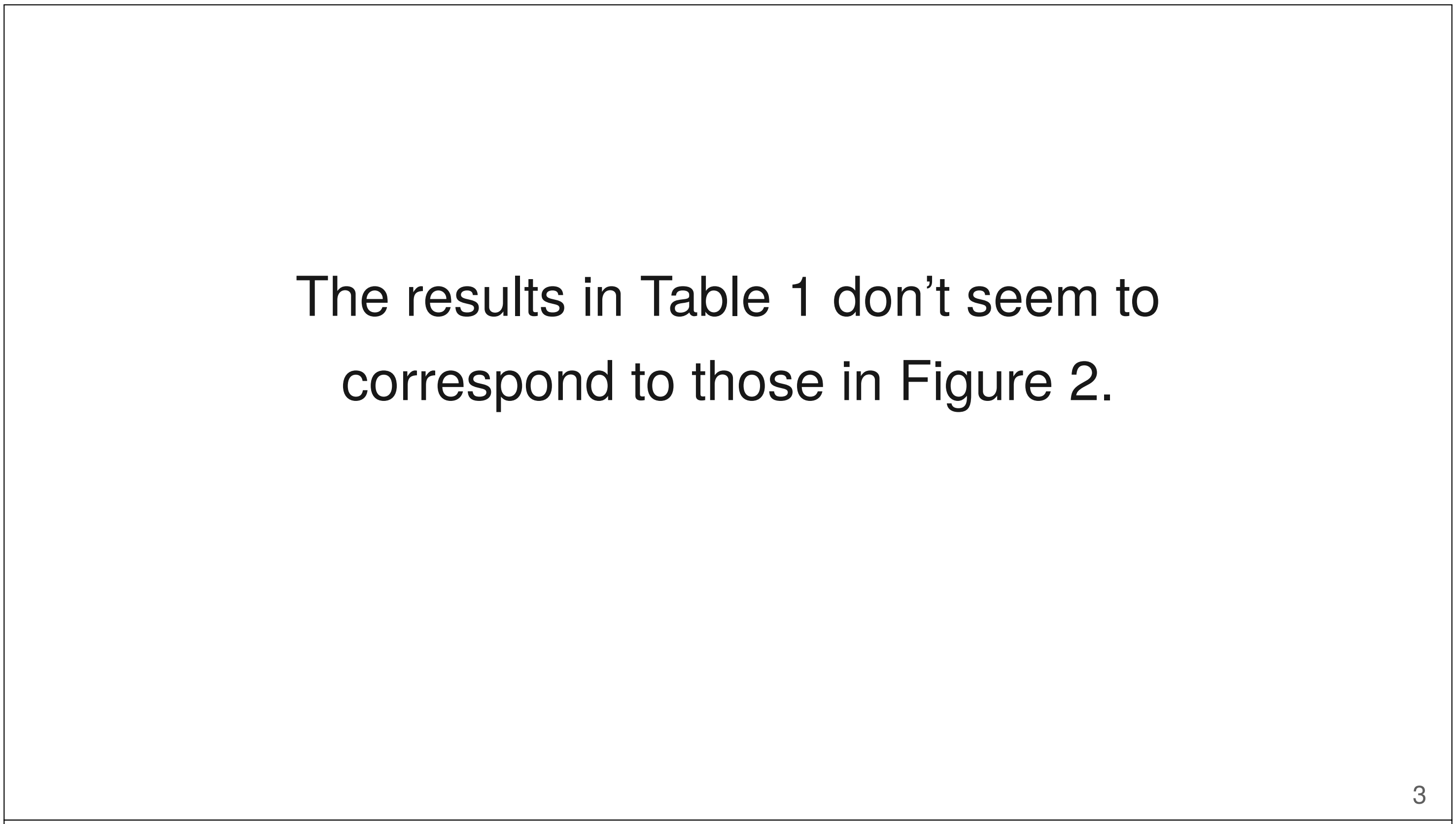
I'm an applied statistician; my goal is to help people make sense of their data. I have a lot of collaborators, and there's nothing I enjoy more than puzzling over their data. So I write a lot of reports, describing what I've done and what I've learned.

This is an email I got from a collaborator, in response to an analysis report that I had sent him. It's always a bit of a shock to get an email like this: what have I done? Why am I working with the wrong data, and where is the right data?

But what he didn't know is that by this point in my life, I'd adopted a reproducible workflow. Because I'd set things up carefully, I could just substitute in the newer dataset, type a single command ("`make`") to rerun the analyses, and get the revised report.

This is a reproducibility success story. We all make mistakes, but if our projects are reproducible, we can nimbly recover from those mistakes.

There is a second important lesson here: At the start of such reports, I always include a paragraph about our shared goals, along with some brief data summaries. By doing so, he immediately saw that I had an old version of the data. If I hadn't done so, we might never have discovered my error.



The results in Table 1 don’t seem to correspond to those in Figure 2.

My computational life is not entirely rosy. This is the sort of email that will freak me out.

## Why did I omit those samples?

I may decide to omit a few samples. Will I record why I omitted those particular samples?

## In what order do I run these scripts?

Sometimes the process of data file manipulation and data cleaning gets spread across a bunch of scripts that need to be executed in a particular order. Will I record this information? Is it obvious what script does what?

“The attached is similar to the code we used.”

From an email in response to my request for code used for a paper.

# Reproducible research

organize the data and code in a way
that you can hand them to someone else
and they can re-run the code
and get the same results
(the same figures and tables)

To reiterate my definition of reproducible research: it's about assembly and organizing the data and code so that they can be re-run to give the same results.

## Steps to reproducible research

- Organize your data and code
- Everything with a script
- Automate the process
- Turn scripts into reproducible reports
- Turn repeated code into functions
- Package functions for reuse
- Use version control
- License your software

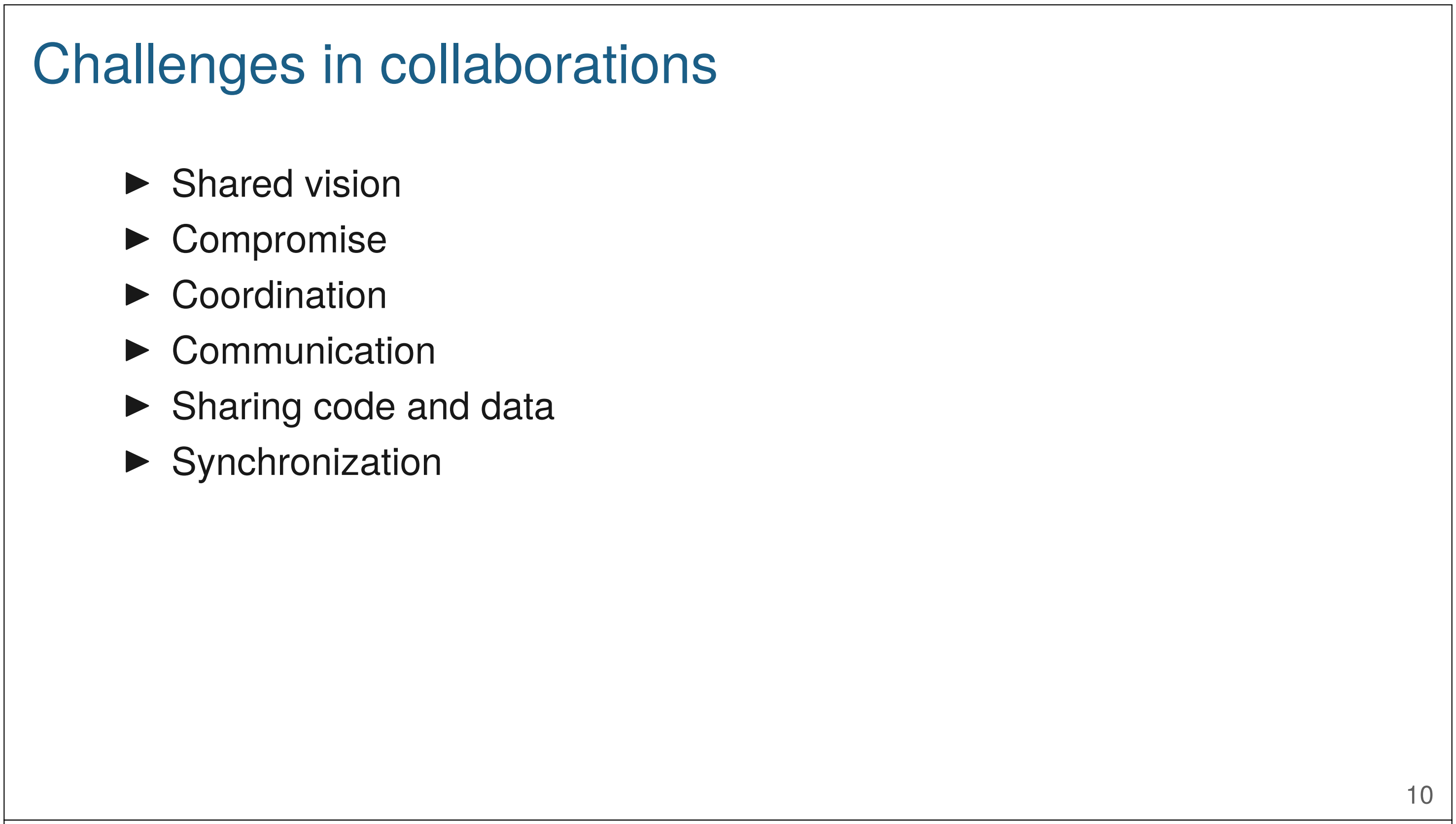
kbroman.org/steps2rr

More than 10 years ago, while preparing for a workshop to develop a course on reproducible research, I thought through the steps one might follow, when transitioning from "standard practice" to a fully reproducible workflow. I created the website kbroman.org/steps2rr.

# Additional considerations

- Arranging data within files
- Metadata
- Sharing data and code
- Software testing
- Capturing the software environment
- Containers
- Handling large-scale computations
- Coordinating with collaborators

There are a number of additional important things that I hadn't covered in kbroman.org/steps2rr.



## Challenges in collaborations

- Shared vision
- Compromise
- Coordination
- Communication
- Sharing code and data
- Synchronization

Collaboration also has challenges.

Do you have a shared vision for the reproducibility of the project? You'll no doubt need to make some compromises about how things are done: you can't both just do things the way you've always done them. Careful coordination and regular communication are key.

And then there are the technical challenges of how to share the code and data and make sure your two working projects remain in sync.

In a sense, the reproducibility of a collaborative project is dependent on the weakest link. If one collaborator refuses to fully participate and share their work, the chain is broken.

# Tools

- R, python
- R Markdown, Quarto, Jupyter notebooks
- GNU make, snakemake, targets
- functions and packages
- git and GitHub
- renv, conda
- docker

There are quite a lot of new tools to learn, when seeking to adopt a reproducible life.

The most important tool is the mindset,
when starting, that the end product
will be reproducible.

– Keith Baggerly

The second-most important tool is training.

So true. Desire for reproducibility is step one.

And I’ve long felt that the key need, in getting computational scientists to adopt a reproducible workflow, is training. For the most part, all of the software tools are available, but many people haven’t incorporated them into their daily work.

Slides: `kbroman.org/Talk_JAXsymp`

`kbroman.org`

`github.com/kbroman`

`@kbroman@rstats.me`

Here's where you can find me, as well as the slides for this talk.