Platforms for Research Collaboration Transparency and Reproducibility in Research

Using **R, Git, and GitHub**

Louis Damian

Why Collaboration Matters in Research

- Modern research is collaborative
- Projects evolve over time
- Multiple contributors, files, and versions
- Poor collaboration leads to:
  - Lost work
  - Confusion
  - Irreproducible results

Common Collaboration Problems

- Emailing files back and forth
- Multiple “final” versions
- No clear ownership of changes
- Difficult to verify results later

What Do We Mean by Transparency?

Transparency refers to:

- Open disclosure of data
- Clear documentation of methods
- Explicit analytic decisions
- Accessible procedures

Purpose:

- Enable understanding
- Enable evaluation
- Enable reuse

What Do We Mean by Reproducibility?

Reproducibility means:

- An independent researcher
- Uses the same data
- Uses the same methods
- Obtains the same results

The Central Question
The key test of reproducible research:

This question guides:

- Tool selection
- Workflow design
- Collaboration choices

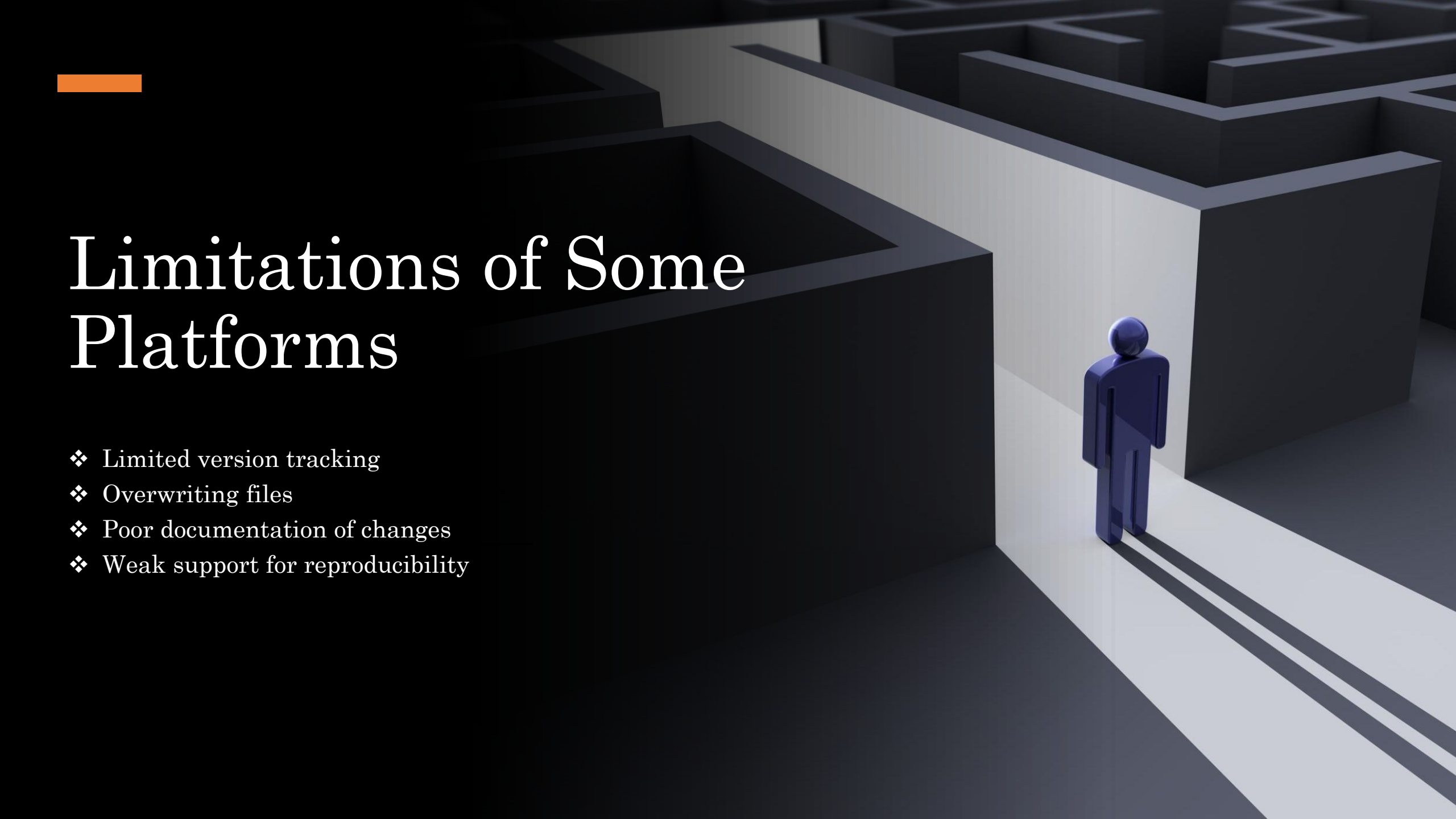
Platforms Used for Research Collaboration

Common platforms include:

- Email
- Google Drive / Dropbox
- Overleaf
- Open Science Framework (OSF)
- GitHub / GitLab

Limitations of Some Platforms

- Limited version tracking
- Overwriting files
- Poor documentation of changes
- Weak support for reproducibility



Why Git and GitHub Stand Out

- Git and GitHub:
  - Track changes over time
  - Record who changed what and when
  - Support collaboration
  - Improve transparency
  - Support reproducible workflows

What Is Git?

- Git is a **version control system**
- It tracks changes in files
- Maintains a full project history
- Allows recovery of earlier versions
- Simple analogy:
  - “Git is a timeline of your research project.”

What Is GitHub?

GitHub is an **online platform** built on Git

- Stores Git repositories
- Enables collaboration
- Makes projects shareable and transparent

Git vs GitHub

Git	GitHub
Tracks changes	Hosts projects
Works locally	Works online
Manages versions	Enables collaboration

How Git Supports Transparency

Git:

- Records every change
- Stores commit messages
- Documents analytical decisions
- Creates a clear research trail

How GitHub Supports Reproducibility

GitHub allows:

- Sharing data and code
- Access to full project history
- Independent verification
- Re-running analyses

Collaboration Using GitHub

GitHub enables:

- Multiple contributors
- Parallel work
- Controlled merging of changes
- Clear contribution records

Why Use Git & GitHub with R?

R projects involve:

- Scripts
- Data
- Outputs

Git + GitHub:

- Track all project files
- Preserve analysis history
- Enable reproducible R workflows

How Git Works with R (Conceptually)

Git does **not** run R code

Git tracks:

- R scripts
- Data files
- Output files

RStudio integrates Git directly

Tools We Will Use (Practical Session)

To work with Git and GitHub in R, we need:

- R
- RStudio
- Git
- GitHub account
- GitHub Desktop (for simplicity)

Linking R, Git, and GitHub

Workflow overview:

- Create an R project
- Track it using Git
- Link it to GitHub
- Collaborate transparently
- Reproduce results easily

Transparency + Reproducibility Workflow

#Data → R Script → Results → Git History → GitHub

Each step is:

- ❖ Documented
- ❖ Shareable
- ❖ Verifiable

What We Will Do Next (Practical Session)

In the next session, you will:

- Install required tools
- Connect RStudio to Git
- Link projects to GitHub
- Collaborate on a project
- Test reproducibility

Key Takeaways

- Transparency shows **how** results were produced
- Reproducibility proves results can be trusted
- Git and GitHub support both
- Simple workflows are powerful

Acknowledgements

This project was funded by the Berkeley Initiative for Transparency in the Social Sciences (BITSS), under the Center for Effective Global Action (CEGA).