portage

Enabling painless reuse of shared research data and code: a case study on computational reproducibility

Ana Trisovic,

Sloan postdoctoral fellow at IQSS,

Harvard University

Qian Zhang,

CLIR postdoc fellow in software curation,

University of Waterloo; Senior Analyst, NDRIO

25/08/2020

Agenda

- Short introduction (5 mins)
- Talk #1 (20 mins):
  - Enabling Painless Reuse of Shared Research Data and Code in data repository Dataverse, by Ana Trisovic
- Talk #2 (20 mins):
  - Enabling Painless Reuse of Shared Research Data and Code for HPC-driven computational reproducibility of research, by Qian Zhang
- Q&A and open discussion (15 mins)

Acknowledgements

- Our research is funded or supported by:
  - Council on Library and Information Resources (CLIR)
  - Portage Network
  - The Alfred P. Sloan Foundation
  - The AWS Cloud Credits for Research program
  - Institute for Quantitative Social Science
  - David R. Cheriton School of Computer Science & University Library, University of Waterloo

Enabling Painless Reuse of Shared Research Data and Code in data repository Dataverse

Ana Trisovic,

Sloan postdoctoral fellow at IQSS,

Harvard University

Agenda of this talk

- Introduction
- Quality of shared data & code
  - How do we ensure it?
- Code execution experiments
  - What happens when we automatically re - execute R or Python code?
  - What are the most common errors?
- Painless research reproducibility and reuse
  - Toward enabling painless reproducibility and reuse in Dataverse

Introduction

- "Reproducibility (computational) is obtaining consistent results using the *same* input data, computational steps, methods and code"
- "Replicability is obtaining consistent results across studies aimed at answering the *same scientific questions*, each of which has obtained its own data"

~ *National Academies of Sciences, Engineering, and Medicine. 2019.* *https://doi.org/10.17226/25303*

Introduction

- Enabling research reproducibility and reuse in practice:
  - Researchers collect, create, process, analyze and interpret data
  - They publish their findings through journal publication
  - They share their data and code (when possible) typically through data repositories
  - Researchers often face numerous degrees of freedom and lack of guidance when sharing data, which later hinder reproducibility and reuse

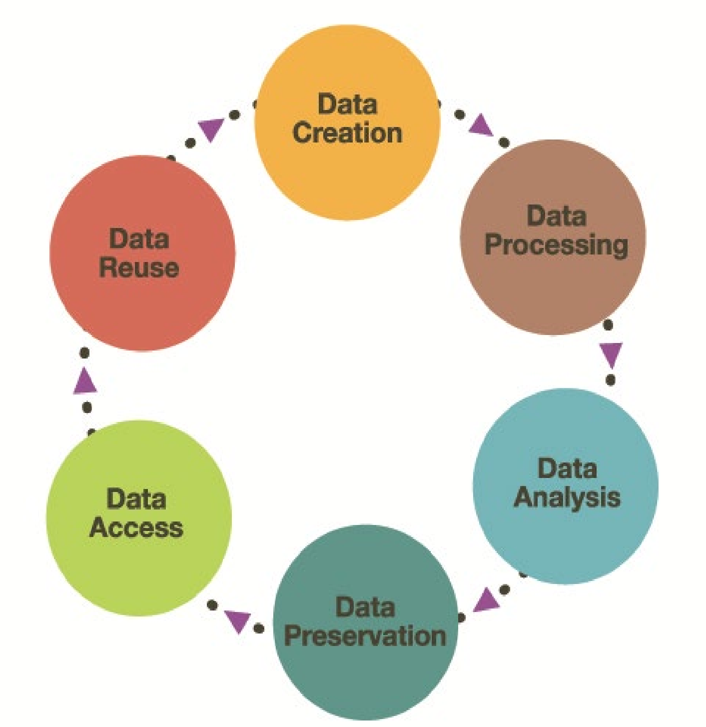


Figure credit: NTU Libraries

Introduction

- Dataverse is a free and open-source software platform to archive, share, and cite research data
- 60 institutions around the globe run Dataverse instances as their data repository

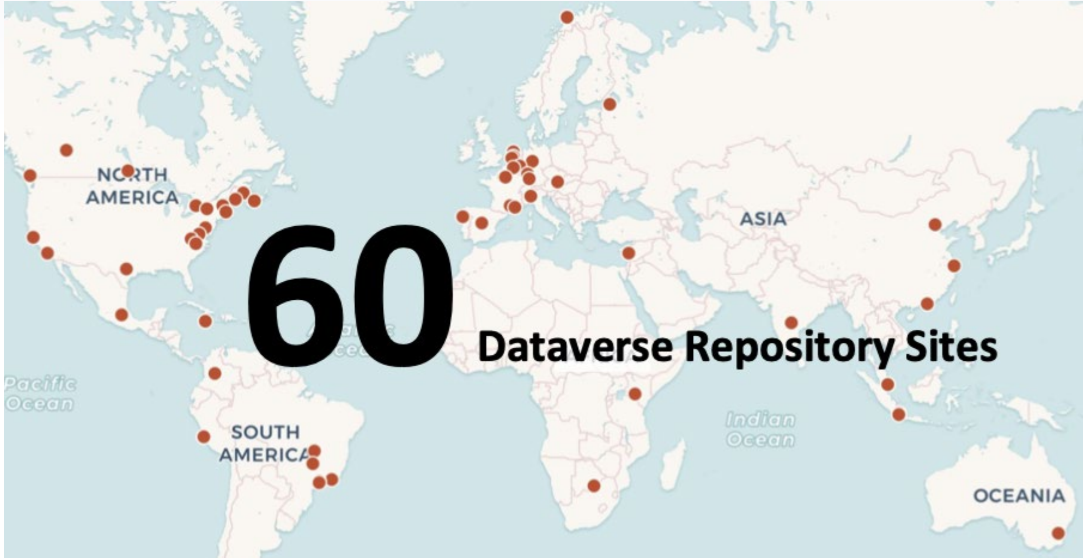
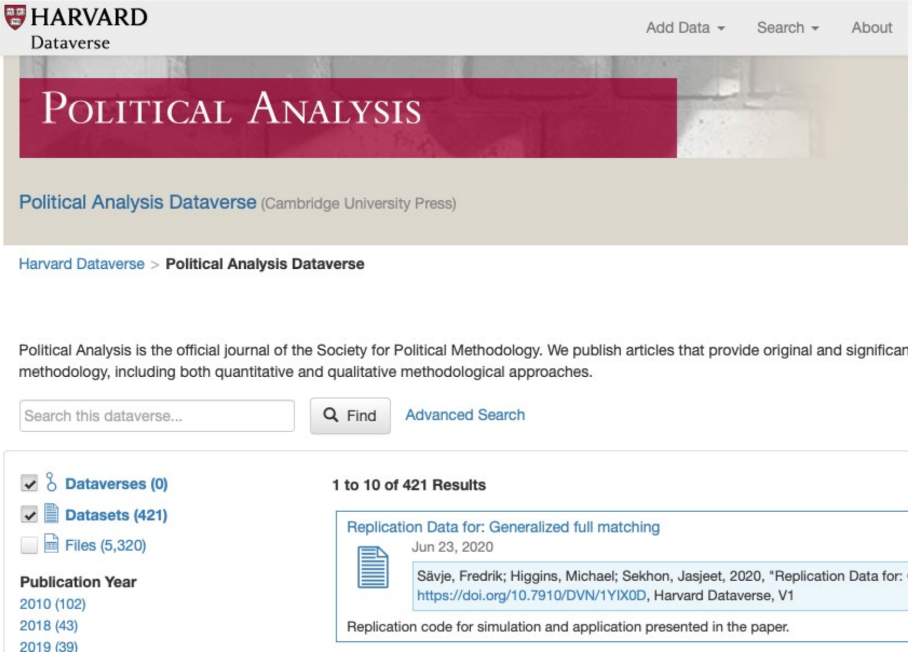
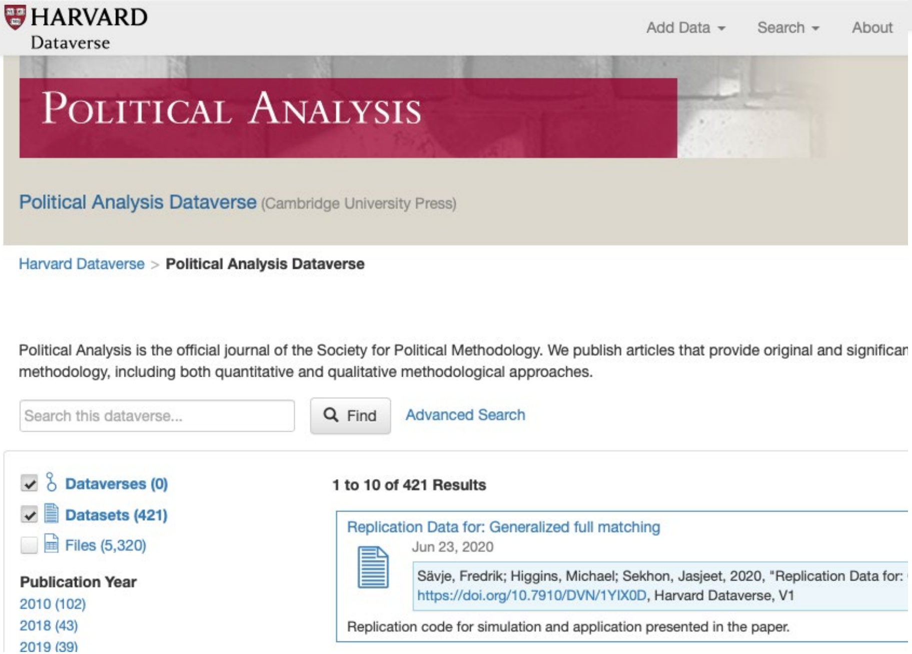


Figure Credit: Merce Crosas

Introduction



Introduction

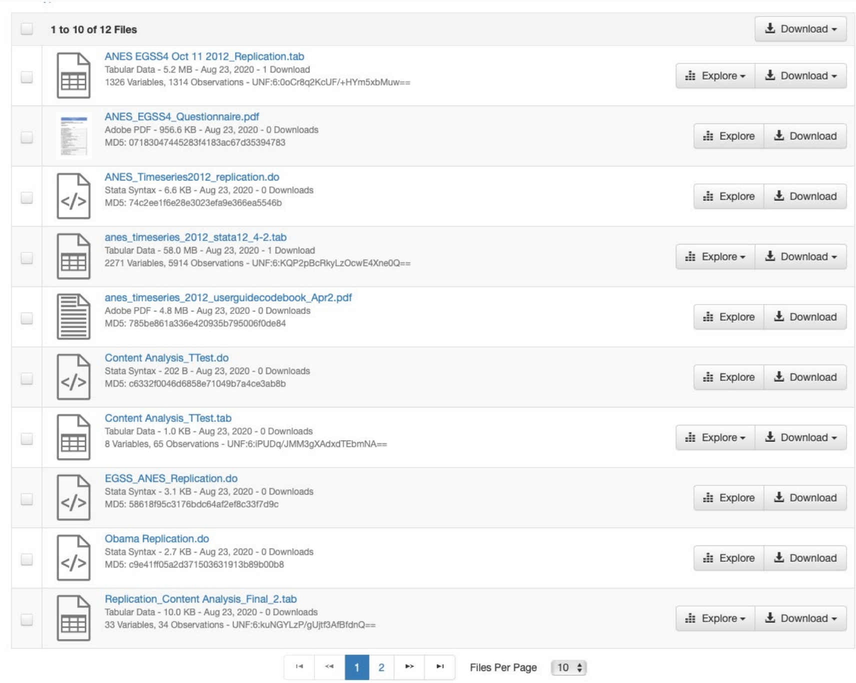


Quality of shared research data & code

- Data quality is determined by its **fitness-for-use** for a given community. Data accuracy, precision, consistency, and completeness are valued across all user communities.
- Before data is published and disseminated, there is a **high potential** in developing its documentation that can improve its fitness for future use.
- After data is deposited, measuring reuse is one way to understand researchers' perceived quality of data products.
  - For example, Harvard Dataverse measures dataset view, downloads, and citations.

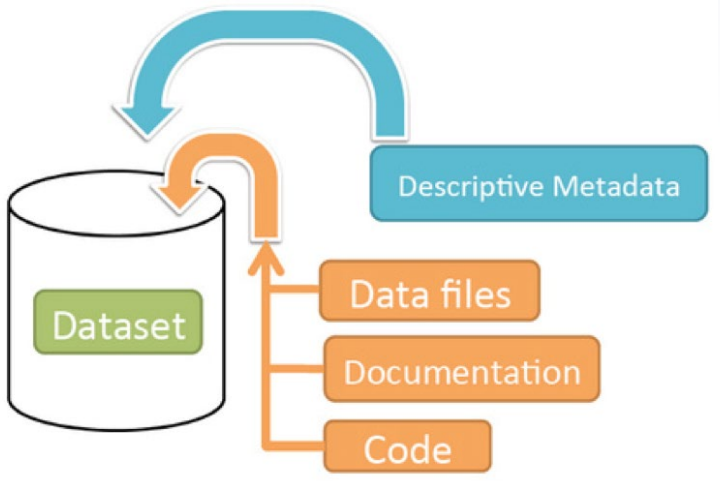
Computational metrics: research code completeness

- Necessary component for reproducibility:
  - Input data
  - Research code
  - Code dependencies (libraries, system dependencies, etc.)
  - Research workflow (i.e., a sequence of analysis steps)
  - Other (computational infrastructure, OS, contextual information etc.)

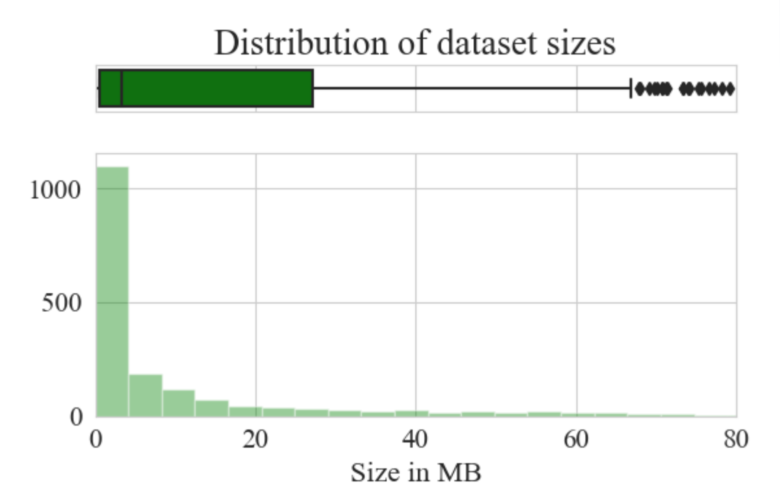


Code execution experiments from Dataverse

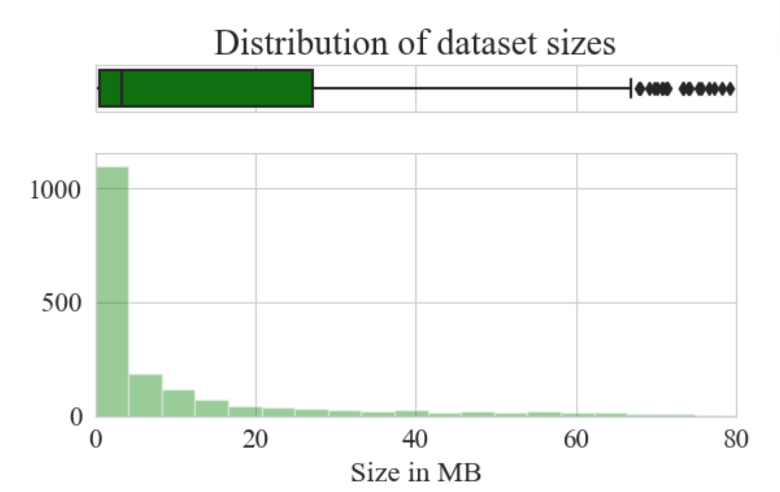
- The experiment is implemented on **AWS Batch**
- A replication dataset contains: R (or Python) code, data and documentation
- Allocated time to run each R file is 1 hour (we also ran experiments with 10 minutes per R file)
- We studied over 2091 datasets, containing over 8178 R files.

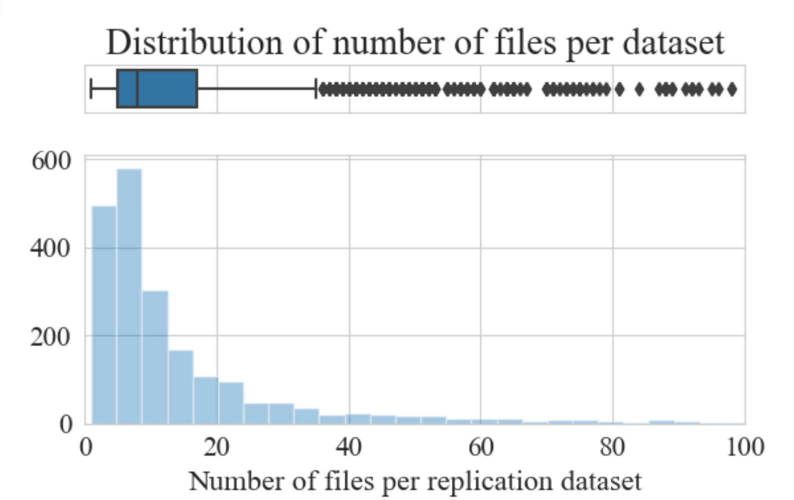


How do datasets with R code look like?

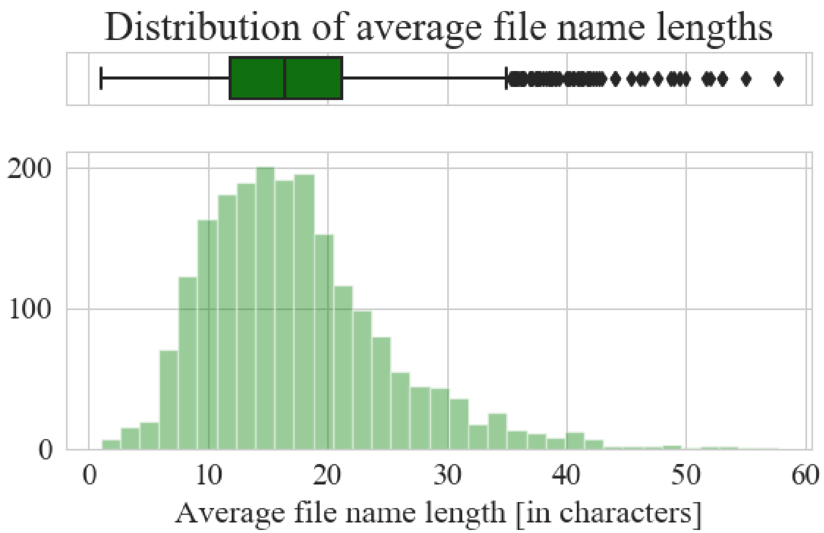


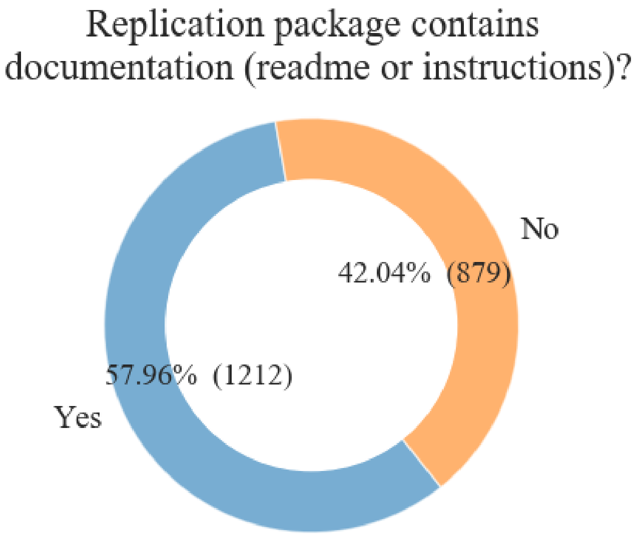
How do datasets with R code look like?





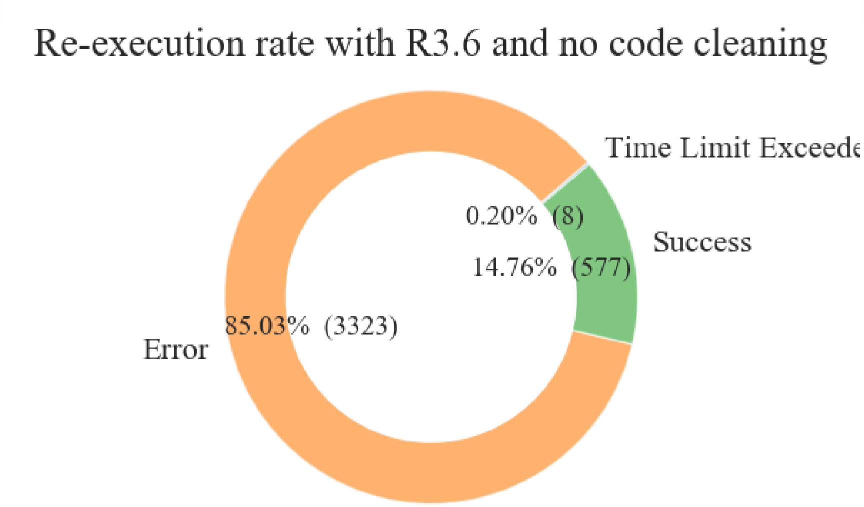
How do datasets with R code look like?





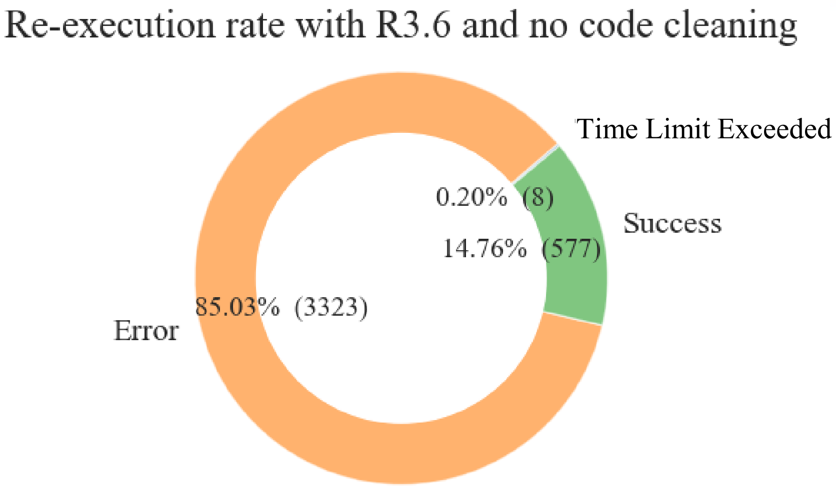
Execution of R code

Without code cleaning:

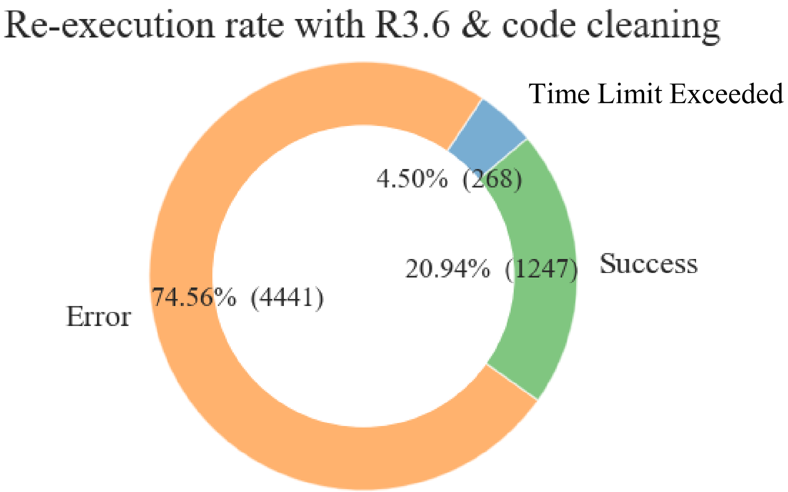


Execution of R code

Without code cleaning:



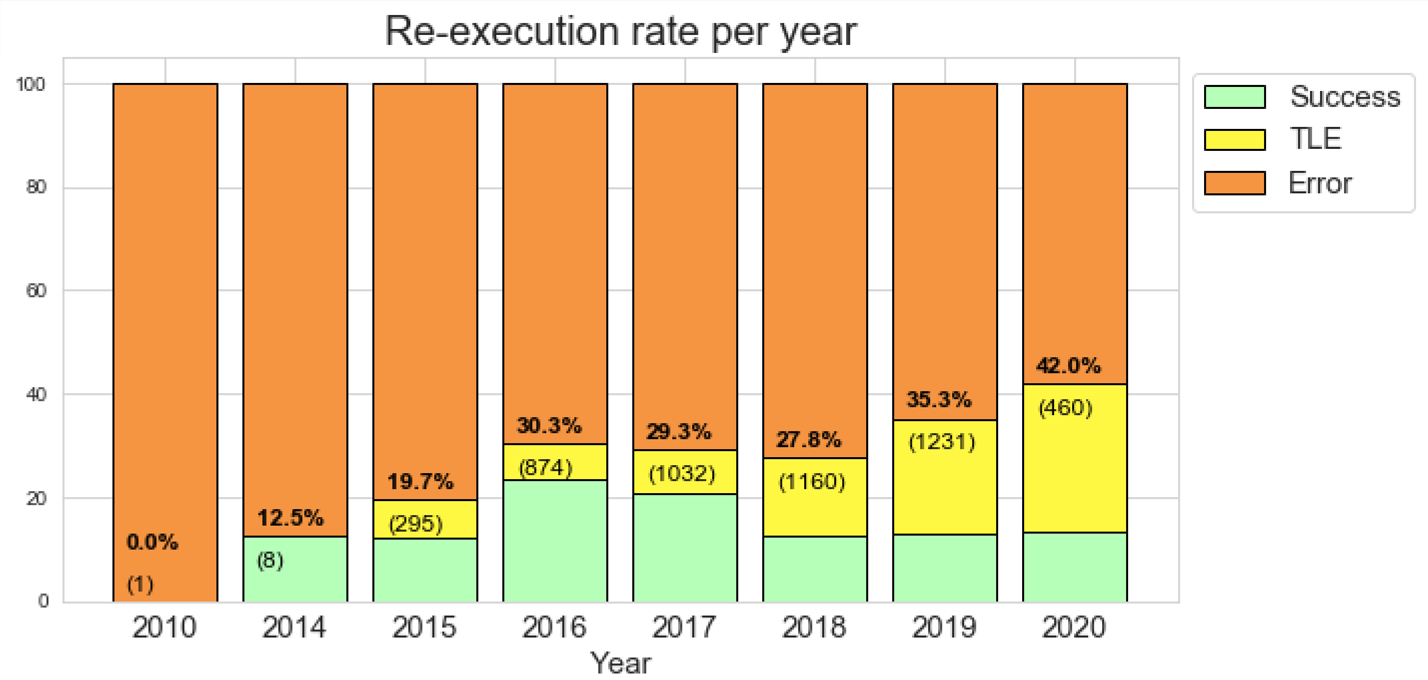
With code cleaning:



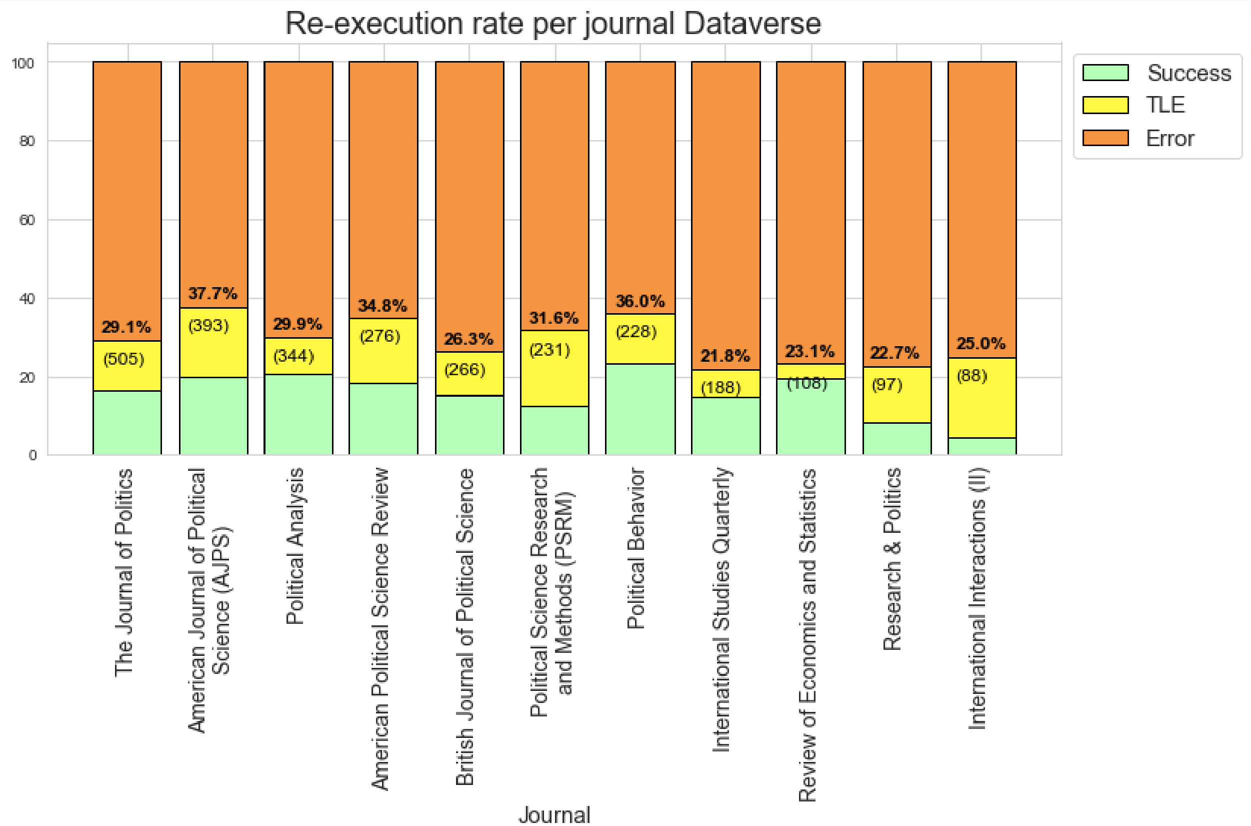
Execution of R code: errors

	Without code cleaning:	With code cleaning:
library	60%	25%
setwd	12%	0%
TLE	1%	15%
file	10%	10%
other	17%	50%

Execution rate of R 3.4 (with code cleaning) files per year of publishing



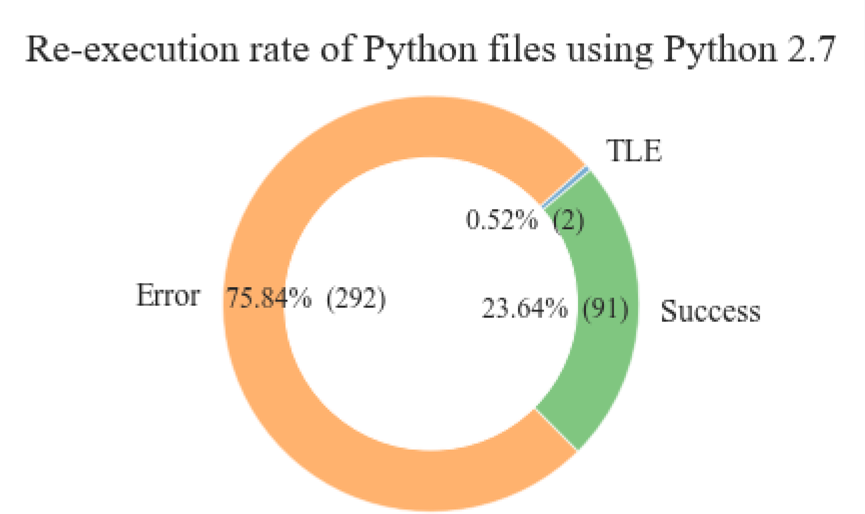
Execution rate per Dataverse journal

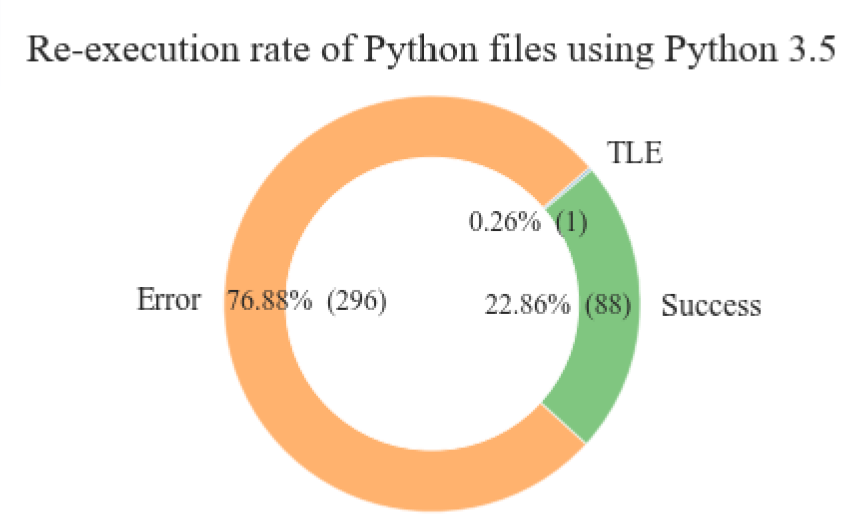


Next steps

- We are analyzing re- execution rate for 3 different versions of R (3.2, 3.6 and 4.0)
  - With varied allocated time for execution (up to 1h per file)
  - Manuscript in preparation
- Also we want to prevent common execution mistakes before depositing code in Dataverse, possibly with an automatic CI (continuous integration)

Execution of Python code with code cleaning

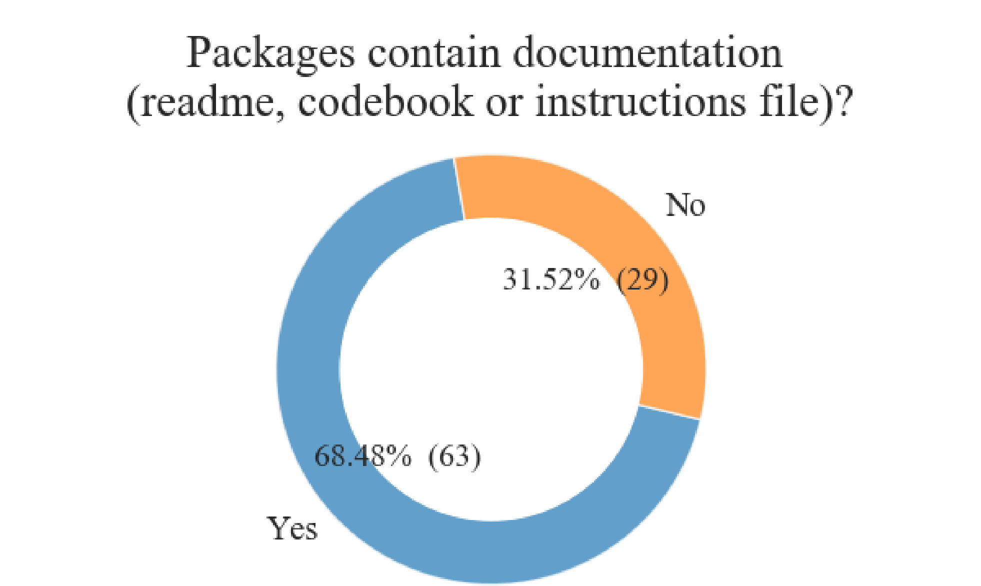




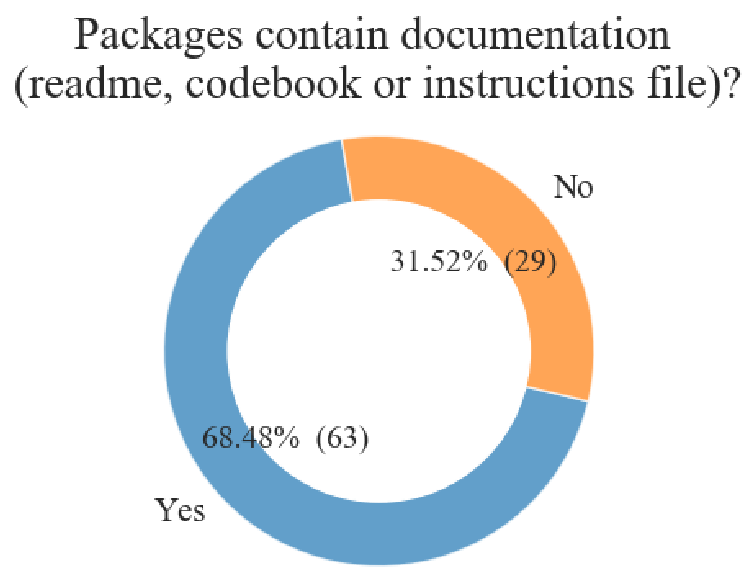
Errors: ImportError, SyntaxError

TLE = time limit exceeded

Datasets with Python code



Datasets with Python code



File	Count (out of 92)
environment.yml	0
requirements.txt	6
Dockerfile	0

Enabling reproducibility and painless reuse

- Container technology (or encapsulation) provides a way to virtualize an OS in a lightweight way and capture data, software and its dependencies
  - It is often used on the cloud
- Containers are becoming popular for preserving research data & code, as they can facilitate research reproducibility and reuse.
  - They are one of the best solutions to enable reproducibility
- There are different types of containers and they can describe research processes in a variety of computing infrastructures

Use of containers in research

- Many new tools encapsulate research data and code in a container "behind the scenes", which capture computational environment that can be shared, reproduced and reused
- Examples:
  - Code Ocean
  - Whole Tale
  - Renku
  - ReproZip
  - Stencila
  - ...

CODE OCEAN

Stencila

ReproZip

WHOLE TALE

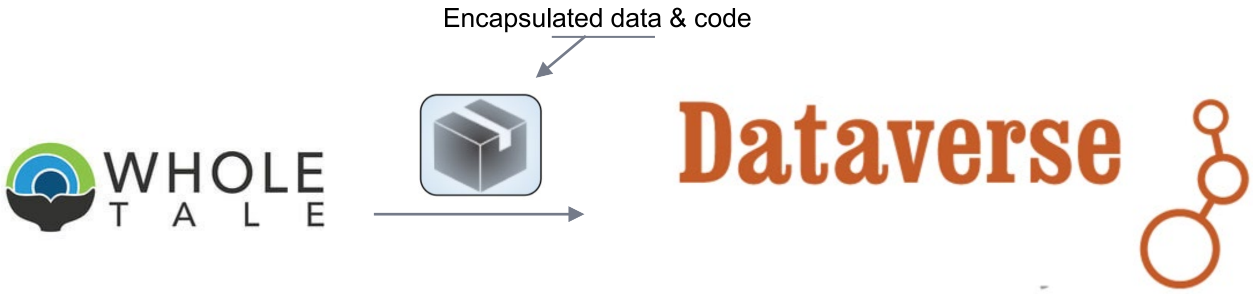
RENKU

Reproducibility platforms vs. data repositories

- Reproducibility platforms support
  - Research portability, reproducibility and reuse
  - However research data and code are not normally findable in data search engines, and there is no commitment for long-term preservation
- Data repositories often support
  - Finability through the use of standard metadata
  - Standardized persistent citation
  - Long-term accessibility of data and code
  - Troubles with enabling reproducibility

Dataverse approach

- Dataverse data repository aims to improve reproducibility of deposited research data & code by developing new functionality to capture containers
  - Ongoing integration with reproducibility platforms Code Ocean, Whole Tale, Renku and Binder, that would allow encapsulated data & code to be exported (deposited) in Dataverse



Dataverse approach

- Dataverse data repository aims to improve reproducibility of deposited research data & code by developing new functionality to capture containers
  - Ongoing integration with reproducibility platforms Code Ocean, Whole Tale, Renku and Binder, that would allow encapsulated data & code to be exported (deposited) in Dataverse
  - Any user should be able to preserve their container based artifacts regardless of their use of the reproducibility platforms.

Outlook

- How to enable painless reuse of shared research data and code?
  - Avoiding common mistakes
  - Including virtual environments in shared code
  - Better metadata to capture ever-more complicated computing infrastructures

Talk #2

Enabling Painless Reuse of Shared Research Data and Code for HPC-driven computational reproducibility of research

Qian Zhang

Agenda of this talk

- What is HPC- driven:
  - computational research?
  - computational reproducibility?
  - Why is it important?
- HPC- driven computational reproducibility: A case study
  - Challenges & Opportunities
- Painless HPC- driven research reproducibility and reuse
- Outlook

What is the HPC-driven computational research?

- *Not theoretical* : deductive mathematics
- *Not experimental* : empirical statistical analysis
- **Computational**: large-scale simulations / data-intensive computational science
  - **Big data**
  - High performance computing (**HPC**): Computational power, application of supercomputers, parallel computing
  - **Software & code** is persuasive in modern digital research landscape

What is the HPC-driven computational reproducibility?

- ⇒ Same research results
  - Different team
  - Same experimental setup
    - Same artifacts
    - Same measurement procedure
    - *Same/different* operating conditions

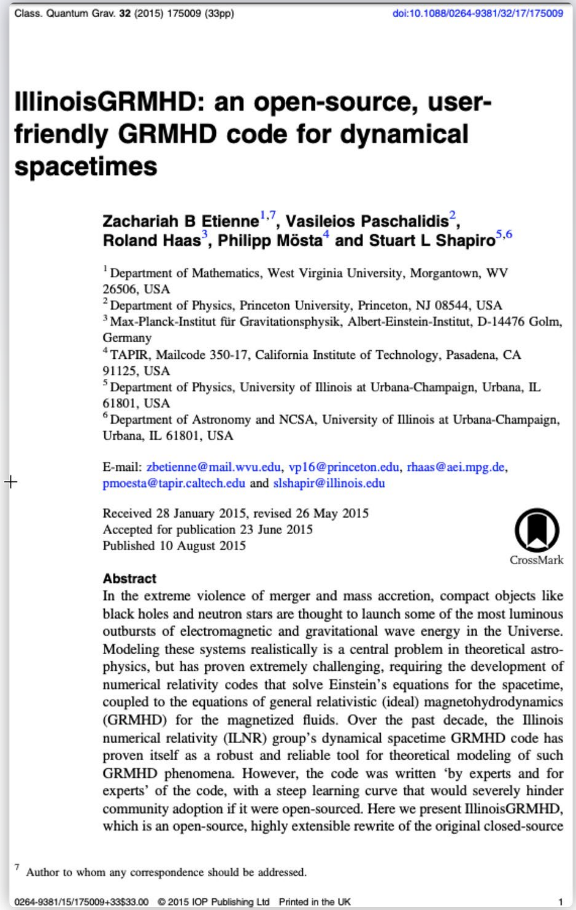
Why does the HPC-driven computational reproducibility?

- "Reproducibility is a Process, not an Achievement" (Lin & Zhang, 2020)
- To root out the error
- Help to "frame the agenda of digital curation" (Stodden, V., 2011. Reproducible Research: A Digital Curation Agenda)
- Central to scientific communication

HPC-driven computational reproducibility: A case study in Astrophysics

- We attempted to reproduce a study:
  - IllinoisGRMHD:

an open - source, user- friendly GRMHD code for dynamical spacetimes (Etienne et al., 2015)


Class. Quantum Grav. 32 (2015) 175009 (33pp) doi:10.1088/0264-9381/32/17/175009

IllinoisGRMHD: an open-source, user-friendly GRMHD code for dynamical spacetimes

Zachariah B Etienne[1,7], Vasileios Paschalidis[2], Roland Haas[3], Philipp Mösta[4] and Stuart L Shapiro[5,6]

[1] Department of Mathematics, West Virginia University, Morgantown, WV 26506, USA
[2] Department of Physics, Princeton University, Princeton, NJ 08544, USA
[3] Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut, D-14476 Golm, Germany
[4] TAPIR, Mailcode 350-17, California Institute of Technology, Pasadena, CA 91125, USA
[5] Department of Physics, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
[6] Department of Astronomy and NCSA, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

E-mail: zbetienne@mail.wvu.edu, vp16@princeton.edu, rhaas@aei.mpg.de, pmoesta@tapir.caltech.edu and slshapir@illinois.edu

Received 28 January 2015, revised 26 May 2015
Accepted for publication 23 June 2015
Published 10 August 2015

CrossMark

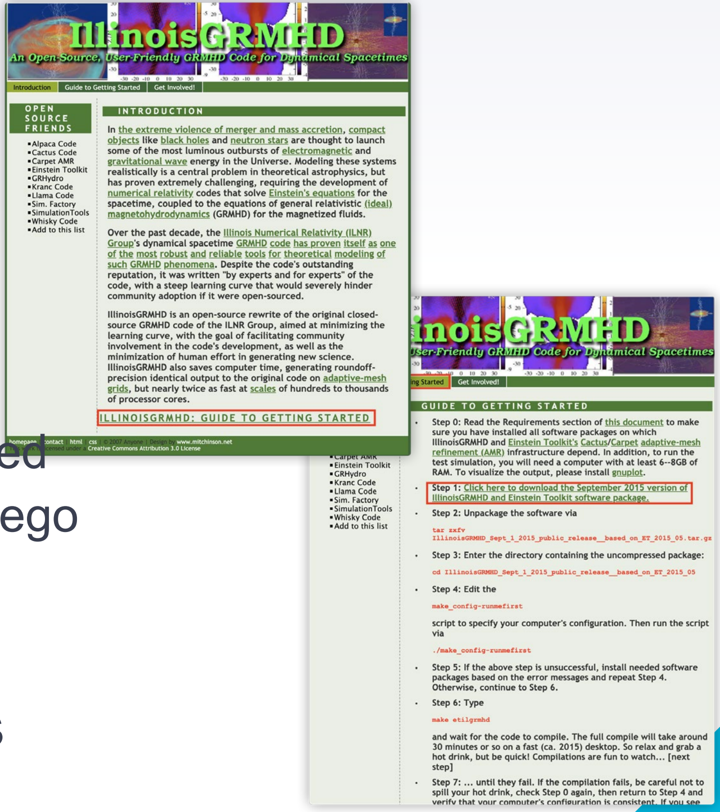
Abstract
In the extreme violence of merger and mass accretion, compact objects like black holes and neutron stars are thought to launch some of the most luminous outbursts of electromagnetic and gravitational wave energy in the Universe. Modeling these systems realistically is a central problem in theoretical astrophysics, but has proven extremely challenging, requiring the development of numerical relativity codes that solve Einstein's equations for the spacetime, coupled to the equations of general relativistic (ideal) magnetohydrodynamics (GRMHD) for the magnetized fluids. Over the past decade, the Illinois numerical relativity (ILNR) group's dynamical spacetime GRMHD code has proven itself as a robust and reliable tool for theoretical modeling of such GRMHD phenomena. However, the code was written 'by experts and for experts' of the code, with a steep learning curve that would severely hinder community adoption if it were open-sourced. Here we present IllinoisGRMHD, which is an open-source, highly extensible rewrite of the original closed-source

[7] Author to whom any correspondence should be addressed.

0264-9381/15/175009+33$33.00

HPC-driven reproducibility setup

- Link to the code: [IllinoisGRMHD]()
  - “Instructions for downloading, compiling, and using IllinoisGRMHD may be found here: http://math.wvu.edu/ ~zetienne/ILGRMHD/”
- HPC resources: XSEDE
  - Stampede2's Skylake (SKX) @Texas Advanced Computing Center (TACC) & Comet @San Diego Supercomputer Center (SDSC)
- Download⇒ compile ⇒ customize the parameter file ⇒ execute ⇒ post-analysis



Preliminary results of the reproducibility case study

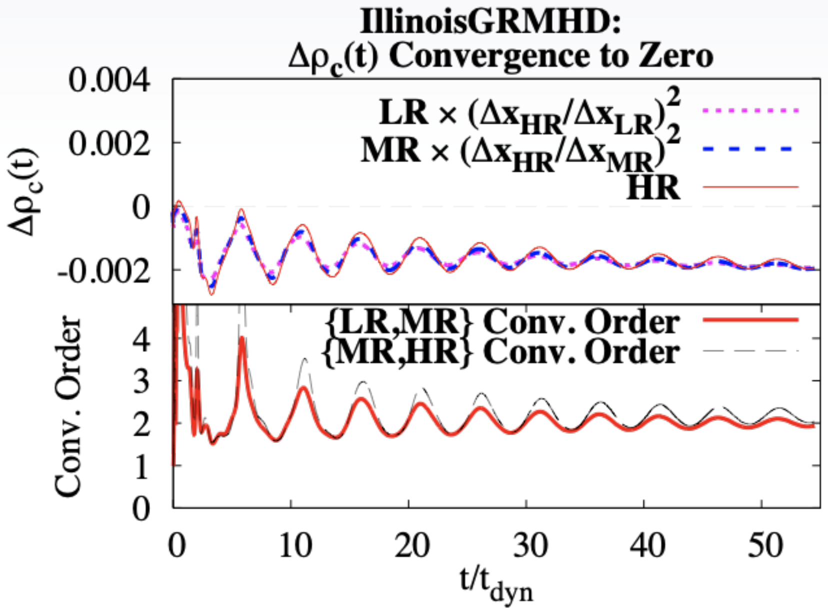
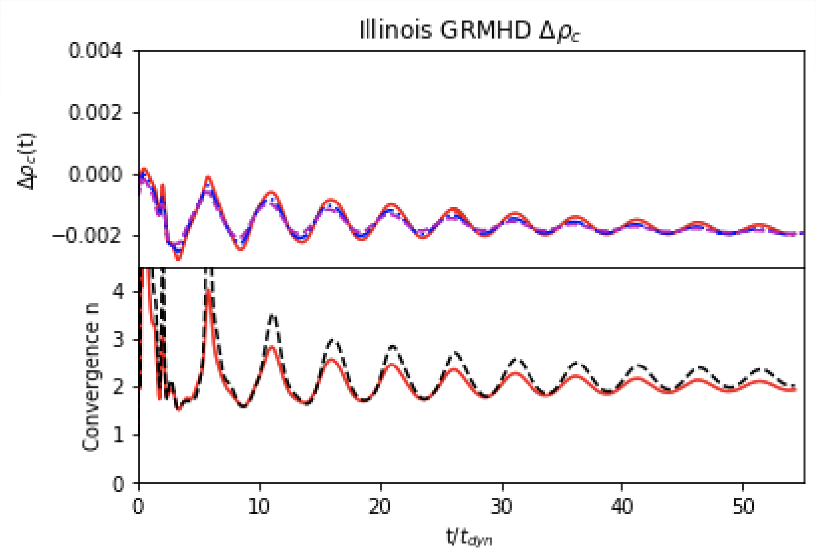


Figure in the paper



Reproducibility experiment result

Observations & lessons learned

- Insufficient data/code
  - Lack of documentation
  - Computational model compilation/execution errors
    - Unstoppable hardware upgrade
    - Link rot, software incompatibilities
  - Missing key parameter (file)

Observations & lessons learned (cont.)

- Installation issues
  - If installed on local laptops
    - Have to be clean slate
  - If installed on local institutional cluster platform
    - ⇒ Setup issues (next slide)
- ⇒ Provide instruction on installation
    - Documentation
    - Checklist

- Issues when submitting jobs (shell script) to queuing system

- ⇒ Provide Human- readable info.

Next step (in progres)

- Develop *generic* setup template to configure a new machine
  - Machine definition
  - Option list: Compilers, Compilation and linking flags, Debugging, Optimisation, Profiling, OpenMP, Warnings, External Libraries (HDF5, MPI, Others)
  - Submission script
  - Run script
- Provide template & examples
  - XSEDE
  - Compute Canada
  - Perimeter Institute for Theoretical Physics
  - AWS

Why are HPC-driven research reproducibility and reuse so difficult?

Why are HPC-driven research reproducibility and reuse so difficult?

- Model
  - Model/code availability/ease of use
  - Platform/system availability
  - Where/how was this run?
  - Model re- usability (setup, etc.)
- Human efforts
- Data
  - Simulation inputs
  - Output usability

Why are HPC-driven research reproducibility and reuse so difficult (cont.)?

- Accessibility
  - Conformance to open or established standards
  - Archival accessibility
  - Longevity of the technology
- Cost
  - Computational cost
  - Storage cost

Opportunities of HPC-driven research reproducibility and reuse

- Ensure **transparency**, **reproducibility** and **reusability** of research results
- Provide effective **communication** of research outputs (publication, data and code) and advanced research computing resources
- Promote enhanced **access** to research outputs and resources
  - Policies and strategies
  - Network and collaborative initiatives
  - Research infrastructures
  - Research software as a primary output of research

Opportunities of HPC-driven research reproducibility and reuse (cont.)

- Develop standards for reproducibility **badges**
  - NISO's Draft Recommended Practice for Reproducibility Badging and Definitions
  - ACM Artifact Review and Badging *Version 1.1 – August 24, 2020*
- **Tools & platforms** for supporting computational science
  - Dissemination/reproducibility platforms (code ocean, Whole Tale)
  - Workflow tracking (Kurator)
  - Better documentation (Jupyter notebook)
- Practices & guidelines
- Training opportunities

https://www.acm.org/binaries/content/gallery/acm/publications/large-replication-badges/all-badges.png

Painless HPC-driven research reproducibility and reuse

- Accessioning, stabilizing, evaluating & describing digital objects
- Documenting and making documentation available
- Sharing resources
  - Data (& documentation) collected & used in analysis
  - Data output result(s) (& documentation) produced by analysis
  - Software (& documentation) in source code & human - readable formats
  - Software/hardware dependencies (technical details, system/software environments)
  - Computational research workflow and provenance
  - Software program(s) dependencies for replicating published results
  - Journal article
- Providing access

Outlook

- Extensive re- use of data and code will become the norm
- Researcher competitiveness will be re-defined with multi-facet metrics
- Cultural change
  - Policy from publishers and funders
  - Author

Takeaways

- "Reproducibility is a Process, not an Achievement"
- Research community's recommendations on good practices
- Greater clarity and guidance on dissemination of computational claims
- Code dissemination in data repositories:
  - Avoiding common mistakes by testing code in a clean environment
  - Including virtual environments in shared code
  - Better metadata to capture ever-more complicated computing infrastructures

Q & A

Questions for the audience

Questions for the audience

- How were your experiences with research reproducibility and reuse? What difficulties have you encountered?
- How do you disseminate data and code at your institution (or research field)? How do you document them?

Thank you for your attention!