Appendix

Reproducibility key to sharing

- Accessible code, e.g. Github
- Accessible, tamper-proof dataset vault, e.g. Arrow
- Accessible environment, e.g. Dockerfiles
- Notebook compiled into computational graph, i.e. shared random seed, externalized jobs
- Verifiable identity, auditable

Notebook Authoring

- VS Code Extension
- Unique in Github or Bitbucket URL per notebook
- Leveraging language servers, etc.
- Add style and interactivity with CSS, Javascript
- Declare runtimes and dependencies

Notebook Readers

- Mobile and Web apps for rendered interactive markdown documents
- Run and experiment with code chunks
- Clone, modify, comment, share

Remote Code Chunks

- Local markdown parser and language servers
- Run R, Python, etc., code chunks on remote sessions
- Local graphics interactivity with Javascript, CSS

Cloud Services

- User management and collaboration services
- Tamper-proof versioned datasets in Apache Parquet
- R, Python, etc., runtimes based on Docker images
- Kubernetes cluster

Remove friction

1. Sign up
2. Install the App
3. Invite collaborators, which can be students
4. Author, find, critique, reuse
5. Share

Execution Graph

- Notebook defines a computation graph

Python Block -> R Block -> R Block

Env in Arrow

Reproduciblity Requires

- Code and data are available
- Toolchain is open source
- Documentation is in plain text
- Data is immutable, inspectable, and described
- Code is versioned and literate
- Dependencies are declared and packaged

More broadly, sharing

- Wikipedia
- Arxiv
- Papers with Code
- GitHub
- DockerHub
- Stack Overflow

Anti examples, more broadly

- Results cannot be replicated
- WYSIWYG documents (Word, Excel) on shared drives
- Sharing by copy, paste
- "It worked on my machine"
- Blackbox models and algorithms
- Automated disinformation
- Spoofing identity and fraud

Authoring Data Science Reproducibly

- Documents built from versioned natural language, code, data, and infrastructure
- Readers can run it, interact with it, and build on it
- Machines can read and exchange it
- Leverage "the Cloud" for scalability
- Identity and assets are verifiable
- Meta authoring

Meta Authoring Supports Usability

- Types and Schemas and "data about data"
- "IntelliSense" and language servers (e.g. VS Code Ecosystem)
- Linked data and entity linking
- Semantic discovery and recommentations (e.g. BERT and GPT2)

Literate Programming

- Markdown, e.g. Knitr
- LaTeX and KaTeX
- Funcitonal Programming
- Grammar of Graphics
- GraphViz
- Dataframes and Pipelines

Language Servers

- Support different IDEs and environments
- Code completion for language and symbols
- Hover, Jump to Ref, Find References
- Diagnostics

Cloud Workloads

- Dockerfiles declare dependencies
- Container as a computational unit
- Kubernetes workloads
- GPUs for vertical scale
- Data in Cloud Storage
- CI/CD Pipelines from Git

Interactive Data Analysis

- Brushing and Linked Data
- Interactive Controls
- Streamlit and Shiny

Sharing and Collaboration

- Built on reproducibility
- Mobile devices
- Export snips
- Share links
- Comment and critique

Markdown Notebook Editing

- Monoco Editor with CSS, Javascript
- R, Python language servers
- iOS, Android clients for phone and tablet
- MS Visual Code Extension for Linux, Mac, Windows Desktop

Interactivity

- Brushing and linking in VegaLite
- Javascript in CSS

Cloud Services

- Remote R, Python, Julia in cloud containers
- Big Data through Spark and Apache Arrow Ballista
- Run GPU containers for RAPIDS, PyTorch, Tensorflow
- Identity Management

Reproducibility and Sharing

- Share notebooks and their cloud runtimes through App
- Github, Bitbucket integration
- Apache Arrow Datasets on Cloud Storage
- Publish documents, e.g. Dropbox

Reading and Sharing

- Notebook has a unique endpoint
- Monoco Editor (from VS Code)
- Native Application
- Manage a set of Notebooks
- Interactive controls for parameters
- Edit mode
- Share with collaborators

Brian Rowe's Presentation and Book

"Achieving Practical Reproducibility with Transparency and Accessibility" (DSSV 2020)

Introduction to Reproducible Science in R

Bio

Soren Harner is a data scientist and CTO with 25+ years in software working in computer vision, natural language, and big data.

Soren Harner Bio Long

Soren is a data scientist working in computer vision, natural language, and big data. In this talk, Soren is representing RC2AI, a company founded by his father, Jim Harner, who passed away earlier this year. In his 25 year career in software, Soren has held executive product leadership roles in companies such as Atlassian, BigCommerce, and MuleSoft. He is currently CTO at LayerJot, a computer vision company active Healthcare. Soren has a MSCS with an AI specialization from Stanford University.

Articles on Reproducibility

- 1,500 scientists lift the lid on reproducibility
- What does research reproducibility mean?
- Trust in science, social consensus and vaccine confidence, Nature 2021 Survey
- What qualifies as scientific authority?

Jim's Software

xStatR UserR2019