

Putting Reproducibility into Practice: Workflows and Case Studies

Rutgers University has made this article freely available. Please share how this access benefits you.
Your story matters. <https://rucore.libraries.rutgers.edu/rutgers-lib/56798/story/>

This work is the **AUTHOR'S ORIGINAL (AO)**

This is the author's original version of a work, which may or may not have been subsequently published. The author accepts full responsibility for the article. Content and layout is as set out by the author.

Citation to *this Version*: Womack, Ryan. *Putting Reproducibility into Practice: Workflows and Case Studies*, 2016.
Retrieved from <http://dx.doi.org/doi:10.7282/T3JS9TR6>.



Terms of Use: Copyright for scholarly resources published in RUcore is retained by the copyright holder. By virtue of its appearance in this open access medium, you are free to use this resource, with proper attribution, in educational and other non-commercial settings. Other uses, such as reproduction or republication, may require the permission of the copyright holder.

Article begins on next page

PUTTING REPRODUCIBILITY INTO PRACTICE: WORKFLOWS AND CASE STUDIES

RUTGERS WORKSHOP ON REPRODUCIBILITY IN
EXPERIMENTAL AND COMPUTATIONAL SCIENCE
RESEARCH

Ryan Womack

Data Librarian, Rutgers University, rwomack@rutgers.edu

October 10, 2016



This work is licensed under a [Creative Commons Attribution
-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

- ▶ I approach reproducibility as a *librarian*
- ▶ Libraries are about *enabling learning*, and building on other's knowledge work
- ▶ Libraries take the long view with respect to keeping and archiving information
- ▶ But we are constantly introducing new users to that information
- ▶ So issues like *usability* and *availability* are paramount

- ▶ What is a *data librarian*?
- ▶ We acquire, locate, and help others use and reuse data
- ▶ So reproducibility concerns are a natural part of my daily work
- ▶ Librarians cheer the larger impacts on science, but others are more qualified to analyze them
- ▶ As librarians, we care about this from the other end
- ▶ We see the frustration when users cannot do what they want with data that is inaccessible or incomprehensible

EXEMPLARS FROM THE SOCIAL SCIENCES

- ▶ **ICPSR** has been in operation for over 50 years, with well-established archiving practices and data documentation via codebooks and metadata
- ▶ **IPUMS** is reformatting and making data compatible across many decades and different projects, to enable international comparisons of microdata
- ▶ Coming from the world in the social sciences where long-term is, if not always routine, at least well-established
- ▶ Disciplinary separation of practices is diminishing when similar computational techniques can be applied to physical sciences or digital humanities

- ▶ We will illustrate some practices in a few contexts
 - ▶ an individual researcher
 - ▶ a team or research group
 - ▶ ongoing, large-scale collaboration

THE DATA IN ITSELF

Some basic practices:

- ▶ Keep raw data pristine and separate from any working data
- ▶ Document your variables and data collection
 - ▶ anything you yourself would forget when revisiting the project 3 years later in response to a query
 - ▶ that will be the same thing other users need too!
- ▶ Don't work in Excel [if you can] or other manual editing environment
 - ▶ you should write down all your steps if you are doing this
 - ▶ better to use code or an environment that will at least record your steps

DOI, the Digital Object identifier, is the great success story

- ▶ makes it easy to have a permanent reference and good citation practice
- ▶ usually associated with quality data repositories
- ▶ encapsulates a lot of good stuff
- ▶ moral: defined standards and centralized tools make adoption and use easy
- ▶ Treat your local data as if you were pulling it from a DOI, and you will be baking in reproducibility

CODE/DATA/DOCUMENT INTEGRATION

We will discuss examples in R, but other programming environments support this as well (Mathematica notebooks, Python/Jupyter)

- ▶ originally implemented in L^AT_EX + Sweave
- ▶ can embed R code and run it as the document is generated
- ▶ Code that is tangled in with text can be extracted, and formatted documents can be woven from the literate program.
- ▶ always ensures that the latest data and results are actually incorporated
- ▶ helps to document and explain code in context (literate programming)
- ▶ PDF, document, and HTML formats are easy to obtain

- ▶ Markdown + [knitr](#) has become a popular, lightweight replacement
- ▶ Simple syntax and implementation
- ▶ Integrated into RStudio
- ▶ Publish documents with one click at [RPods](#)
- ▶ Can fall back on \LaTeX /Sweave for more complex document formatting

A SHORT RMD FILE

```
— title: "A short literate programming exercise" author: "Ryan Womack"
date: "October 10, 2016" output: pdf_document —
“{r setup, include=FALSE} knitr::opts_chunk$set(echo = TRUE) “
## Read in the data
Let's read in the data with the following commands:
“{r load} library(readxl)
download.file("http://ryanwomack.com/data/PharmaDemo.xls",
"mydata.xls")
mydata<-read_excel("mydata.xls")
names(mydata)
attach(mydata)
“
## Describe the Data
Then we will get some summary statistics on the Age and Weight variables:
“{r summary} summary(Age)
summary(Weight) “
Now plot the data:
“{r plot, echo=FALSE} library(ggplot2)
ggplot(mydata, aes(Weight, Age))+ geom_point() “
## Regression
“{r regression} summary(lm(Age~Weight))
ggplot(mydata, aes(Weight, Age))+ geom_point()+ stat_smooth() “
All done!
```

RSTUDIO ENVIRONMENT

Putting
Reproducibility
into Practice:
Workflows and
Case Studies

The screenshot displays the RStudio interface with three main panes:

- Source Editor:** Contains R code for data analysis:

```
30
31 Then we will get some summary statistics on the Age and Weight variables:
32
33 (r summary)
34 summary(Age)
35
36 summary(Weight)
37
38
39 Now plot the data:
40
41 (r plot, echo=FALSE)
42 library(ggplot2)
43
44 ggplot(mydata, aes(Weight, Age))+ geom_point()
45
46
47 ## Regression
48
49 (r regression)
50 summary(lm(Age~Weight))
51
52 ggplot(mydata, aes(Weight, Age))+ geom_point()+ stat_smooth()
53
54
55 All done!
```
- Console:** Shows the output of the R code:

```
summary(Age)
  Min.   10  Median   30   Max.
-24.626 -5.943  1.847  6.411 19.980

summary(Weight)
  Min. 1st Qu.  Median  Mean 3rd Qu.  Max.
 44.00  74.00  90.40  90.87 105.70 139.00

Coefficients:
(Intercept) 83.00577    2.89586 28.664 < 2e-16 ***
Weight      -0.18489    0.03108 -5.949 1.21e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.056 on 198 degrees of freedom
Multiple R-squared:  0.1516, Adjusted R-squared:  0.1473
F-statistic: 35.39 on 1 and 198 DF, p-value: 1.205e-08

> getwd()
[1] "/home/ryan/R"
> setwd("/home/ryan/wosack/documents/ryan/work/2016/")
>
```
- Environment/History:** Displays a literate programming document titled "A short literate programming exercise" by Ryan Wosack, dated October 10, 2016. The document includes:
 - Read in the data:** Code to download data from a URL and read it into R, followed by a data summary table:

name	mydata
## [1]	"Age"
## [2]	"Weight"
## [3]	"Sex"
## [4]	"Height"
## [5]	"BMI"
## [6]	"Hypertension"
## [7]	"Diabetes"
## [8]	"Cholesterol"
## [9]	"Glucose"
## [10]	"Insulin"
## [11]	"C-peptide"
## [12]	"A1C"
## [13]	"HbA1c"
## [14]	"HbA1c_1C"
## [15]	"HbA1c_2C"
## [16]	"HbA1c_3C"
## [17]	"HbA1c_4C"
## [18]	"HbA1c_5C"
## [19]	"HbA1c_6C"
## [20]	"HbA1c_7C"
## [21]	"HbA1c_8C"
## [22]	"HbA1c_9C"
## [23]	"HbA1c_10C"
## [24]	"HbA1c_11C"
## [25]	"HbA1c_12C"
## [26]	"HbA1c_13C"
## [27]	"HbA1c_14C"
## [28]	"HbA1c_15C"
## [29]	"HbA1c_16C"
## [30]	"HbA1c_17C"
## [31]	"HbA1c_18C"
## [32]	"HbA1c_19C"
## [33]	"HbA1c_20C"
## [34]	"HbA1c_21C"
## [35]	"HbA1c_22C"
## [36]	"HbA1c_23C"
## [37]	"HbA1c_24C"
## [38]	"HbA1c_25C"
## [39]	"HbA1c_26C"
## [40]	"HbA1c_27C"
## [41]	"HbA1c_28C"
## [42]	"HbA1c_29C"
## [43]	"HbA1c_30C"
## [44]	"HbA1c_31C"
## [45]	"HbA1c_32C"
## [46]	"HbA1c_33C"
## [47]	"HbA1c_34C"
## [48]	"HbA1c_35C"
## [49]	"HbA1c_36C"
## [50]	"HbA1c_37C"
## [51]	"HbA1c_38C"
## [52]	"HbA1c_39C"
## [53]	"HbA1c_40C"
## [54]	"HbA1c_41C"
## [55]	"HbA1c_42C"
## [56]	"HbA1c_43C"
## [57]	"HbA1c_44C"
## [58]	"HbA1c_45C"
## [59]	"HbA1c_46C"
## [60]	"HbA1c_47C"
## [61]	"HbA1c_48C"
## [62]	"HbA1c_49C"
## [63]	"HbA1c_50C"
## [64]	"HbA1c_51C"
## [65]	"HbA1c_52C"
## [66]	"HbA1c_53C"
## [67]	"HbA1c_54C"
## [68]	"HbA1c_55C"
## [69]	"HbA1c_56C"
## [70]	"HbA1c_57C"
## [71]	"HbA1c_58C"
## [72]	"HbA1c_59C"
## [73]	"HbA1c_60C"
## [74]	"HbA1c_61C"
## [75]	"HbA1c_62C"
## [76]	"HbA1c_63C"
## [77]	"HbA1c_64C"
## [78]	"HbA1c_65C"
## [79]	"HbA1c_66C"
## [80]	"HbA1c_67C"
## [81]	"HbA1c_68C"
## [82]	"HbA1c_69C"
## [83]	"HbA1c_70C"
## [84]	"HbA1c_71C"
## [85]	"HbA1c_72C"
## [86]	"HbA1c_73C"
## [87]	"HbA1c_74C"
## [88]	"HbA1c_75C"
## [89]	"HbA1c_76C"
## [90]	"HbA1c_77C"
## [91]	"HbA1c_78C"
## [92]	"HbA1c_79C"
## [93]	"HbA1c_80C"
## [94]	"HbA1c_81C"
## [95]	"HbA1c_82C"
## [96]	"HbA1c_83C"
## [97]	"HbA1c_84C"
## [98]	"HbA1c_85C"
## [99]	"HbA1c_86C"
## [100]	"HbA1c_87C"
## [101]	"HbA1c_88C"
## [102]	"HbA1c_89C"
## [103]	"HbA1c_90C"
## [104]	"HbA1c_91C"
## [105]	"HbA1c_92C"
## [106]	"HbA1c_93C"
## [107]	"HbA1c_94C"
## [108]	"HbA1c_95C"
## [109]	"HbA1c_96C"
## [110]	"HbA1c_97C"
## [111]	"HbA1c_98C"
## [112]	"HbA1c_99C"
## [113]	"HbA1c_100C"
 - Describe the Data:** Code to generate summary statistics for Age and Weight:

```
summary(Age)
##      Min. 1st Qu.  Median  Mean 3rd Qu.  Max.
## 23.0  40.0  46.0  46.0  56.0  89.0

summary(Weight)
##      Min. 1st Qu.  Median  Mean 3rd Qu.  Max.
## 44.00  74.00  90.40  90.87 105.70 139.00
```
 - Now plot the data:** A scatter plot showing Age (y-axis) versus Weight (x-axis). The plot shows a positive correlation between weight and age, with a density gradient from light to dark.

THE COMPUTING ENVIRONMENT

- ▶ Open source is an important enabler of reproducibility
- ▶ Anyone can grab copies of the software to execute
- ▶ And can get older versions if necessary for compatibility
- ▶ You can also record information about your computing environment (`session.Info()` in R)
- ▶ The [checkpoint](#) package automates this process in R

- ▶ Don't save output. Where did it come from? This should be done in the code.
- ▶ Clean, well-formatted data (tidyr) and code (formatR) are a plus
- ▶ If using README files approach, document everything

- ▶ **Jupyter** grew out of iPython
 - ▶ now over 40 languages supported
- ▶ Mathematica Notebooks, cloud support
- ▶ Cloud services making sharing much easier
- ▶ Becoming an expectation

COLLABORATION

- ▶ The same forces (cloud computing, shared platforms, standards) are making collaboration easier than ever
- ▶ [Github](#), [Bitbucket](#), and others enable easy collaboration on programming
 - ▶ with significant side benefits for reproducibility due to availability of code
- ▶ The [Open Science Framework](#) provides a more data-specific approach
- ▶ A key feature is that the same platform is used for private work and then public sharing
- ▶ [Psychology reproducibility study](#) uses OSF.
 - ▶ See the [Science article](#) for a start

LIKELIHOOD YOU WILL GET CODE WORKING
BASED ON HOW YOU'RE SUPPOSED TO INSTALL IT:



► <http://xkcd.com/1742/>

- ▶ Collaborative projects must/should agree on:
 - ▶ data practices and sharing platform
 - ▶ coding practices and sharing platform
- ▶ This is made easier by already existing platforms and practices

The [Yale Institute for Social and Policy Studies](#) is an example of a research group that enables reproducibility.

- ▶ Data and papers archived together onsite
- ▶ Handles (not DOIs) for data
- ▶ Code and documentation archived
- ▶ Code review for correct execution
- ▶ Good example of providing explanatory metadata for studies
- ▶ Possible because the Institute requires compliance as a condition of grants

Multi-year, multi-institutional projects that may continue beyond original PIs *require* reproducibility.

- ▶ Many people will be coming on and off of the project over time
- ▶ Many unanticipated uses are anticipated (“known unknowns” or something like that)
- ▶ Collaboration and continuity must be consciously planned for
- ▶ Decisions should be made with more consideration for future use than current convenience
- ▶ But disciplinary expertise is building in these areas
- ▶ PDB, of course
- ▶ Also, <https://www.teamsciencetoolkit.cancer.gov/>

Introduction

Individuals
(Everyone)

Collaboration

Team Science

Conclusion

- ▶ In big science, the main node(s) are enablers of future reuse
- ▶ They provide basic infrastructure (OOI)
- ▶ But also provide a clearinghouse for other projects that link to and build on the central node
- ▶ One major future goal is to have more generic, all-purpose collaborative infrastructure
- ▶ Rutgers is developing a Virtual Data Collaboratory for this purpose
- ▶ Open infrastructure like [Zenodo](#), [Dataverse](#), [OpenICSPR](#) and the [Open Science Data Cloud](#) have been developed

BIG SCIENCE - PRACTICES

- ▶ Standards such as DOI and ORCID enable the broader community to coalesce around good practice
- ▶ Data repositories are developing standards
- ▶ Re3data.org is a directory of repositories
- ▶ The [Data Seal of Approval](#) is awarded to repositories using sound data practices
- ▶ [ISO 16363](#) (Trusted Digital Repositories) is a more stringent standard
- ▶ One important step is to plan for what happens when the project winds down (expectedly or unexpectedly)
- ▶ What is the equivalent standard for computing and reproducibility?

BIG SCIENCE - GENOMIC DATA SHARING

Putting
Reproducibility
into Practice:
Workflows and
Case Studies

Ryan Womack

- ▶ Massive investments, massive amounts of data
- ▶ Many repositories too (NIH GDS)
- ▶ Existing repositories are useful and aggregate many software tools
- ▶ But researchers want even larger pooled databases, especially for human genome
- ▶ Technical issues are complex, but the rights and permissions involved are equally complex
- ▶ How can data be federated for maximum discoverability?

Introduction

Individuals
(Everyone)

Collaboration

Team Science

Conclusion

OPEN DATA -> REPRODUCIBLE RESEARCH

Putting
Reproducibility
into Practice:
Workflows and
Case Studies

Ryan Womack

- ▶ Increasing openness is a long-term trend
 - ▶ Internet, Government, Data, Software, Cloud Computing
- ▶ Pressure for Reproducible Research can only increase as these trends intensify
- ▶ Good news is...
 - ▶ Benefits are clear for society and for knowledge creation!
 - ▶ Tools to enable this are getting easier all the time!
 - ▶ Eventually it will be a standard we will take for granted!

Introduction

Individuals
(Everyone)

Collaboration

Team Science

Conclusion