Reproducible Data Analysis

Al Cooper

Presentation to the EOL Science Group Nov 2020

* Important Disclaimer: I'm not an expert in the coding aspects that I will discuss. I present this in the hope that you will find this approach interesting and informative.

---

* This presentation is available at this URL:
https://github.com/WilliamCooper/ReproducibleResearch/blob/master/ReproducibleResearchPresentation.pdf

Outline

REPRODUCIBLE RESEARCH vs REPRODUCIBLE ANALYSIS

The Distinction

- Broad view:
  - Scientific progress relies on experiments that test theory.
  - The "gold standard" is replication of results.
  - That's not always possible:
    Weather conditions, for example, are unique.
- "Reproducible Reporting" is different:

> *Given the experimental conditions and data:*
> *Make the report and analysis reproducible.*

THE ADVANTAGES OF REPRODUCIBLE RESULTS

THE ADVANTAGES OF REPRODUCIBLE RESULTS

- Convincing readers and reviewers: Even if no one replicates the results, knowing that one could increases confidence in the results.

THE ADVANTAGES OF REPRODUCIBLE RESULTS

- Convincing readers and reviewers: Even if no one replicates the results, knowing that one could increases confidence in the results.
- Extending results: Others (or you) may want to apply the analysis to new situations.

THE ADVANTAGES OF REPRODUCIBLE RESULTS

- Convincing readers and reviewers: Even if no one replicates the results, knowing that one could increases confidence in the results.
- Extending results: Others (or you) may want to apply the analysis to new situations.
- Meeting journal requirements: Increasingly, journals encourage or even require this.

THE ADVANTAGES OF REPRODUCIBLE RESULTS

- Convincing readers and reviewers: Even if no one replicates the results, knowing that one could increases confidence in the results.
- Extending results: Others (or you) may want to apply the analysis to new situations.
- Meeting journal requirements: Increasingly, journals encourage or even require this.
- Addressing questions that may arise years later that require you or others to return to the project.

THE ADVANTAGES OF REPRODUCIBLE RESULTS

- Convincing readers and reviewers: Even if no one replicates the results, knowing that one could increases confidence in the results.
- Extending results: Others (or you) may want to apply the analysis to new situations.
- Meeting journal requirements: Increasingly, journals encourage or even require this.
- Addressing questions that may arise years later that require you or others to return to the project.
- Assisting collaborators: Teams working together know the details of each member's contributions and can check their work.

WHAT DOES "REPRODUCIBILITY" INVOLVE?

The idea is to make it possible for someone else (or you) to repeat the steps leading to a documented result.

Minimum Requirements:

1. "Provenance" of data used in the analysis:
   (a) How were data obtained?
   (b) Where can data be obtained?
2. Archive all the relevant files in curated repositories:
   (a) any programs used in data analysis (see, e.g., zenodo)
   (b) the figures and how they were generated;
   (c) special modifications to the data files.
3. Favor "DOI" repositories because they will remain unchanged.
4. Where appropriate, provide supplementary guidance in a "Workflow" document.
5. Record the status of the system used (software versions, etc.) –> [See "docker", not discussed further here.]

AN EXAMPLE FROM A RECENT PROJECT

Project:	SensibleHeatFluxTechNote
Archive package:	SensibleHeatFluxTechNote.zip
Contains:	attachment list below
Program:	SensibleHeatFluxTechNote.Rnw
Original Data:	[UCAR/NCAR - EOL(2011)]
Special Data Files:	SensibleHeatFluxTechNote.Rdata
Workflow Document:	WorkflowSensibleHeatFlux.pdf
Git:	https://github.com/WilliamCooper/

attachments:

SensibleHeatFluxTechNote.Rnw
SensibleHeatFluxTechNote.pdf
WorkflowSensibleHeatFlux.pdf
SensibleHeatFluxTechNote.Rdata
SensibleHeatFlux.bib
*chunks/**
SessionInfo

AN EXAMPLE FROM A RECENT PROJECT

Project:	SensibleHeatFluxTechNote
Archive package:	SensibleHeatFluxTechNote.zip
Contains:	attachment list below
Program:	SensibleHeatFluxTechNote.Rnw
Original Data:	[UCAR
Special Data Files:	Sensib
Workflow Document:	WorkflowSensibleHeatFlux.pdf
Git:	https://github.com/WilliamCooper/

The program is Rnw format, mixing LaTeX and R code.

attachments:

SensibleHeatFluxTechNote.Rnw
SensibleHeatFluxTechNote.pdf
WorkflowSensibleHeatFlux.pdf
SensibleHeatFluxTechNote.Rdata
SensibleHeatFlux.bib
*chunks/**
SessionInfo

AN EXAMPLE FROM A RECENT PROJECT

Project:	SensibleHeatFluxTechNote
Archive package:	SensibleHeatFluxTechNote.zip
Contains:	attachment list below
Program:	Sensib
Original Data:	[UCAF
Special Data Files:	Sensib
Workflow Document:	WorkflowSensibleHeatFlux.pdf
Git:	https://github.com/WilliamCooper/

A .zip file contains everything needed to repeat the project. It is saved on GitHub.

attachments:

SensibleHeatFluxTechNote.Rnw
SensibleHeatFluxTechNote.pdf
WorkflowSensibleHeatFlux.pdf
SensibleHeatFluxTechNote.Rdata
SensibleHeatFlux.bib
*chunks/**
SessionInfo

AN EXAMPLE FROM A RECENT PROJECT

Project:	SensibleHeatFluxTechNote
Archive package:	SensibleHeatFluxTechNote.zip
Contains:	attachment list below
Program:	SensibleHeatFluxTechNote.Rnw
Original Data:	[UCAR/NCAR - EOL(2011)]
Special Data Files:	SensibleHeatFluxTechNote.Rdata
Workflow Document:	WorkflowSensibleHeatFlux.pdf
Git:	https://github.com/WilliamCooper/

attachments:

SensibleHeatFluxTechNote.Rnw
SensibleHeatFluxTechNote.pdf
WorkflowSensibleHeatFlux.pdf
SensibleHeatFluxTechNote.Rd
SensibleHeatFlux.bib
*chunks/**
SessionInfo

A Workflow document contains extra info too detailed for the main document

AN EXAMPLE FROM A RECENT PROJECT

Project:	SensibleHeatFluxTechNote
Archive package:	SensibleHeatFluxTechNote.zip
Contains:	attachment list below
Program:	SensibleHeatFluxTechNote.Rnw
Original Data:	[UCAR/NCAR - EOL(2011)]
Special Data Files:	SensibleHeatFluxTechNote.Rdata
Workflow Document:	WorkflowSensibleHeatFlux.pdf
Git:	https://github.com/WilliamCooper/

attachments:

SensibleHeatFluxTechNote.Rnw
SensibleHeatFluxTechNote.pdf
WorkflowSensibleHeatFlux.pdf
SensibleHeatFluxTechNote.Rdata
SensibleHeatFlux.bib
*chunks/**
SessionInfo

SessionInfo includes a detailed list of the components (like R packages) and their version numbers.

TOOLS FOR THIS PURPOSE

There are many options:

1. Just archive everything:
   (a) text files
   (b) programs
   (c) figures and how they were generated
   (d) etc.
2. Use one of the tools developed for this purpose:
   (a) "notebooks" (iPython, Jupyter, Markdown, ...)
   (b) MATLAB "Live Editor"
   (c) "LyX" – supports inclusion of text and R code in the same file.
   (d) R Markdown with "knitr" – My emphasis today.

R Markdown (.Rmd)

Reasons to consider this:

R Markdown (.Rmd)

Reasons to consider this:

1. Generating text with R Markdown is mostly intuitive and easily learned, yet flexible. (Much easier than LaTeX, which is used in .Rnw files.)

R Markdown (.Rmd)

Reasons to consider this:

1. Generating text with R Markdown is mostly intuitive and easily learned, yet flexible. (Much easier than LaTeX, which is used in .Rnw files.)
2. You can include program snippets (chunks) intermixed with text, so that the code does appropriate data processing and generates figures and other results.

R Markdown (.Rmd)

Reasons to consider this:

1. Generating text with R Markdown is mostly intuitive and easily learned, yet flexible. (Much easier than LaTeX, which is used in .Rnw files.)
2. You can include program snippets (chunks) intermixed with text, so that the code does appropriate data processing and generates figures and other results.
3. Code can be incorporated from many different programming languages, including R, Python, FORTRAN, C, C++, and many others (including shell commands).

R Markdown (.Rmd)

Reasons to consider this:

1. Generating text with R Markdown is mostly intuitive and easily learned, yet flexible. (Much easier than LaTeX, which is used in .Rnw files.)
2. You can include program snippets (chunks) intermixed with text, so that the code does appropriate data processing and generates figures and other results.
3. Code can be incorporated from many different programming languages, including R, Python, FORTRAN, C, C++, and many others (including shell commands).
4. While easiest in RStudio, R Markdown only requires R. It can be used on EOL computers.

R Markdown (.Rmd)

Reasons to consider this:

1. Generating text with R Markdown is mostly intuitive and easily learned, yet flexible. (Much easier than LaTeX, which is used in .Rnw files.)
2. You can include program snippets (chunks) intermixed with text, so that the code does appropriate data processing and generates figures and other results.
3. Code can be incorporated from many different programming languages, including R, Python, FORTRAN, C, C++, and many others (including shell commands).
4. While easiest in RStudio, R Markdown only requires R. It can be used on EOL computers.
5. Many output formats are supported: HTML, PDF, LaTeX, Word, LibreOffice, and others.

R Markdown (.Rmd)

Reasons to consider this:

1. Generating text with R Markdown is mostly intuitive and easily learned, yet flexible. (Much easier than LaTeX, which is used in .Rnw files.)
2. You can include program snippets (chunks) intermixed with text, so that the code does appropriate data processing and generates figures and other results.
3. Code can be incorporated from many different programming languages, including R, Python, FORTRAN, C, C++, and many others (including shell commands).
4. While easiest in RStudio, R Markdown only requires R. It can be used on EOL computers.
5. Many output formats are supported: HTML, PDF, LaTeX, Word, LibreOffice, and others.
6. Copernicus journals (like Atmospheric Measurement Techniques) accept submissions in .Rmd format.

ILLUSTRATIONS (all available at URLs at the end)

Some very simple cases:

1. Using text only.
2. Using an R program "chunk".
3. Using a Python program chunk.
4. Plotting a weather map with discussion.

More complex cases:

1. Using both R and Python.
2. Passing variables from R to Python and from Python to R.
3. Incorporating a plot produced by ncplot.
4. A data-quality memo.
5. A journal article.

THE BARE TEMPLATE FROM RSTUDIO:

```
---
title: 'Example 1: Text Only'
author: "Al Cooper"
date: "10/27/2020"
output: html_document
---
## R Markdown
This is an R Markdown document. Markdown is a simple
formatting syntax for authoring HTML, PDF, and
MS Word documents. For more details on using R Markdown
see <http://rmarkdown.rstudio.com>.
In RStudio, click the **Knit** button to generate
a document that includes both content as well as the
output of any embedded R code chunks within the
document. From R, use *rmarkdown::render(path_to_file)*.
```

THE RESULT: Example 1

Note in the output:

- The html code is here and the R Markdown file is here.
- The header is formatted to display at the top of the page.
- "##" produces a second-level heading.
- "<link>" produces a link to a web page
- "**text** causes the text to be bold; *text* italicizes

Additional information in "Example1.html":

- To the above text, I added some introductory examples to illustrate formatting options including bold, italic, superscripts, subscripts, headings, ietc.
- That example also includes a simple way to include an existing image into the document.

Example 2: Include R code

Examples of R-code: a setup "chunk"

```
```{r, setup, include=FALSE, echo = TRUE}
library(knitr)
knitr::opts_chunk$set(echo = TRUE, include = TRUE)
library(Ranadu)
library(reticulate) ## Needed to mix python and R
```
```

- The sequences of back-tic marks denote the start and end of the code chunk.
- This chunk just loads some R packages that will be used later.
- "include = FALSE" keeps error messages from appearing in the output.

Example 2: Include R code

Examples of R-code: get data and make a plot"

```
"'{r, rChunk1, include = TRUE, echo = TRUE}
DF <- getNetCDF(setFileName('SOCRATES', 'rf15'))
DF <- selectTime(DF, 50000, 60000)
## get some statistics:
statsATX <- lapply(WACf, function(f) f(DF$ATX))
statsDPXC <- lapply(WACf, function(f) f(DF$DPXC))
nm <- names(statsATX) ## save the names
ggplotWAC(DF[, c('Time', 'ATX', 'DPXC')])
"'
```

- "getNetCDF()" constructs a data.frame from a netCDF file.
- The "%>%" pipes results to the next statement, here imposing restrictions on the time interval and variables.
- "lapply" applies several statistical functions like "mean()" to the variables, producing named lists.
- "ggplotWAC(DF)" constructs a plot from the data.frame.

Example 3: Using Python code

Setting up to use Python with R Markdown:

1. As far as I can tell, R markdown works only with Python3.

Example 3: Using Python code

Setting up to use Python with R Markdown:

1. As far as I can tell, R markdown works only with Python3.
2. Instructions for Python set-up are provided by RStudio at this link, so I won't be covering that: https://support.rstudio.com/hc/en-us/articles/360023654474-Installing-and-Configuring-Python-with-RStudio.

Example 3: Using Python code

Setting up to use Python with R Markdown:

1. As far as I can tell, R markdown works only with Python3.
2. Instructions for Python set-up are provided by RStudio at this link, so I won't be covering that: https://support.rstudio.com/hc/en-us/articles/360023654474-Installing-and-Configuring-Python-with-RStudio.
3. In addition to standard Python packages (numpy, pandas, matplotlib) you will probably want netcdf4. Confusingly, the package is installed as netcdf4 but you have to import it as netCDF4 (with CDF capitalized).

Example 3: Using Python code

Setting up to use Python with R Markdown:

1. As far as I can tell, R markdown works only with Python3.
2. Instructions for Python set-up are provided by RStudio at this link, so I won't be covering that: https://support.rstudio.com/hc/en-us/articles/360023654474-Installing-and-Configuring-Python-with-RStudio.
3. In addition to standard Python packages (numpy, pandas, matplotlib) you will probably want netcdf4. Confusingly, the package is installed as netcdf4 but you have to import it as netCDF4 (with CDF capitalized).
4. You need the R packages "reticulate", "rmarkdown" and "knitr".

Example 3: Using Python code

Setting up to use Python with R Markdown:

1. As far as I can tell, R markdown works only with Python3.
2. Instructions for Python set-up are provided by RStudio at this link, so I won't be covering that: https://support.rstudio.com/hc/en-us/articles/360023654474-Installing-and-Configuring-Python-with-RStudio.
3. In addition to standard Python packages (numpy, pandas, matplotlib) you will probably want netcdf4. Confusingly, the package is installed as netcdf4 but you have to import it as netCDF4 (with CDF capitalized).
4. You need the R packages "reticulate", "rmarkdown" and "knitr".
5. I will be showing some examples on EOL computers, but most of this presentation is prepared on my home computer where I have RStudio available. That is by far the easiest way to do this work.

Example 3, continued: Using Python code

Making a plot using Python:

1. Include an "R" code chunk to load needed libraries (probably just knitr and reticulate)
2. In a Python code chunk, import packages you need to use, here netCDF4 and matplotlib:

```
from matplotlib import pyplot as plt
import netCDF4 as nc
```

3. Add another Python chunk to load the data and generate a plot:

```
## see example3.html
```

4. Example3 also illustrates use of a "Pandas" data.frame and some ways to improve the appearance of the plot.

Example 4: Inserting an externally generated figure

See Example4.html

- The example includes a simple download of a web file (a weather map).
- A plot generated externally is also included, with some suggestions regarding how to ensure reproducibility.

RStudioAndPython.html

1. This example includes some of the material presented here at a web address that is globally accessible.
2. Some added aspects include:
   (a) some setup guidance
   (b) how to use R code and Python code together
   (c) how to transfer variables from R to Python and v.v.
   (d) making R data.frames from Pandas DataFrames and v.v.

A DATA-QUALITY MEMO

Finding a functional relationship predicting the cavity pressure in dewpoint hygrometers

The problem: During data review, determine if the pressure in the cavity is as expected from past projects.

The approach: Fit data from earlier projects to find equations predicting the measured cavity pressure.

The memo: See this link

The Rmd file: See this link

Note: Results from the fit are incorporated into the text without any need to copy them.

A JOURNAL ARTICLE: Sensible-Heat Flux

Components of the Rmd file:

- https://github.com/WilliamCooper/ SensibleHeatFlux/ blob/master/ SensibleHeatFluxAMT.Rmd
- This was prepared for *Atmospheric Measurement Techniques* and follows a template that they provide.
- Some important components:
  1. The YAML header: (as provided by Copernicus and filled in)
     (a) Includes sections for the abstract, acknowledgements and bibliography, which get placed into the text appropriately.
     (b) Has a section "availability" where you can include reproducibility information.
  2. I have used LaTeX code for equations; there are many other ways to generate these.
  3. Processing via rmarkdown::render() produces a .pdf file and also a .tex file suitable for submission to the journal.
- See the draft manuscript here: https://github.com/WilliamCooper/ SensibleHeatFlux/ blob/master/ SensibleHeatFluxAMT.pdf

SUMMARY

Reproducibility in Data Analysis

Increases confidence in results.

Documents for the future and for extensions.

Incorporates scientific ethics – a strong defense against research misconduct.

Will be increasingly important as studies grow in complexity.

Not just for journals:

- Data quality reports.
- Documentation of processing programs and algorithms.
- Technical notes.
- ...

Many new supporting tools are now available.
I have emphasized R Markdown, but
there are many other ways to do this.

END OF THIS PRESENTATION

Here are links to the programs and resulting documents discussed in this talk:

R Markdown File	Output File
Example1.Rmd	Example1.html
Example2.Rmd	Example2.html
Example3.Rmd	Example3.html
Example4.Rmd	Example4.html
RStudioAndPython.Rmd	RStudioAndPython.html
FitCavityPressure.Rmd	FitCavityPressure.pdf
SensibleHeatFluxAMT.Rmd	SensibleHeatFluxAMT.pdf

This presentation is available at this URL:
https://github.com/WilliamCooper/ReproducibleResearch/blob/master/ReproducibleResearchPresentation.pdf