Eating Your My Own Dog Food

Lessons learned from teaching researchers best practices in computing and data management
About Me

Christina Koch
Research Computing Facilitator
Core Computational Technology

curious
philosophical
sanguine
organizer
bibliophile
Research Computing Facilitator

Center for High Throughput Computing; (also Advanced Computing Initiative, Wisconsin Institute for Discovery)

Interface between researchers and computational experts

Communicate researcher needs to computing “providers”

Engage with researchers to help them leverage (large-scale) computing to support and transform their work
Research Computing Facilitator

Center for High Throughput Computing; (also Advanced Computing Initiative, Wisconsin Institute for Discovery)

...I also teach workshops.
Software and Data Carpentry

International non-profit organizations that facilitate workshops on software development and data skills for research
We offer Software Carpentry and Data Carpentry workshops at UW Madison.

Typical workshop content includes:

- **SWC**: using the unix shell, version control, effective use of programming concepts like loops, functions, and naming conventions.
- **DC**: cleaning data, using programmatic tools like SQL and R to analyze and visualize data.

Workshop content is rooted in **best practices** for software development or data management, as applied to research.
What's the big deal about best practices?
The Wildly Functional Studio of Video Wizard Casey Neistat

https://vimeo.com/68558518
“It's just pure functionality;

I mean, there’s so much s*** in here; all of it’s used, otherwise it wouldn’t be in here, and you have to keep track of it.

So every minute that I spend looking for, like, velcro tape, or my drill gun, is a minute I’m not doing something productive so why not build an infrastructure that supports that rather than encourages the chaos?”
Best practices: ways of thinking and acting that make you more effective.
Objectives For Today

Storytelling
- Share some of my own experience: big picture ideas and best practices from Software and Data Carpentry that I have found helpful in my own work

Adoption
- Find a new tool or idea that you could use to improve your work

Application
- Connect this “best practices” thinking with your own processes
1. Be kind to future you
“Your closest collaborator is you six months ago, but you don’t reply to emails.”

Apocryphal quote from the Software/Data Carpentry community
Measure *twice* once, cut once

Use folders! Organize projects!

- project/
- data/
- scripts/
- figures/

---

@naupakaz on getting organized: "This is so slow! I haven't touched the data yet! I have to make the graph! ..."

6:43 AM - 11 Jan 2016

Believe me, in three months, you will be so happy you did this." #dcuwm

6:44 AM - 11 Jan 2016
Measure **twice** once, cut once

Use folders! Organize projects!
- project/
- data/
- scripts/
- figures/

Write things down!
- Project documentation
- READMEs for EVERYTHING.
Measure **twice** once, cut once

Use folders! Organize projects!

- project/
- data/
- scripts/
- figures/

Write things down!
- Project documentation
- READMEs for EVERYTHING.

Use sensible names!

**NO**
- myabstract.docx
- Joe’s Filenames Use Spaces and Punctuation.xlsx
- figure 1.png
- fig 2.png
- JW7d^(2sl@deletethisandyourcareerisoverWx2*.txt

**YES**
- 2014-06-08_abstract-for-sla.docx
- joes-filenames-are-getting-better.xlsx
- fig01_scatterplot-talk-length-vs-interest.png
- fig02_histogram-talk-attendance.png
- 1986-01-28_raw-data-from-challenger-o-rings.txt
Also, version control

Version control has many faces.

Ultimately it should allow you to:

Track changes

Who made them

And be able to reverse them

Bonus: have development branches
2. Automate *all* the things
My rule of thumb:

If you are

1) copying and pasting
2) doing a lot of clicking

There is probably something you can automate.
Examples of automation

- Sending lots of emails → Mail merge or email templating
- Running the same series of commands over and over → Scripts
- Using the same piece of code in different places → Functions
- Repeated procedure → Checklist
- Answering the same old user question → Documentation
Principles of automation

- Write each thing once
  - Use functions for repeated tasks
  - Use variables for repeated values
  - Write disjoint tools that can be used together

- Designing functions:
  - What varies? What stays the same?
  - Varying values become your input
  - Input → Black Box function → Output
3. Structure Your Data
Data Carpentry: "Anything is possible if you structure your data."
Software Carpentry: "Anything is possible if you structure your code."

Greg Wilson, founder of Software Carpentry
https://twitter.com/gvwilson/status/764543640299528192
What is data?

Data I see on a regular basis:

- Reports of usage hours
- Time tracking
- Office hour attendance
- New accounts
Data is powerful!
Structuring data

Structured data allows you to automate analysis

Different ways to structure data:
- Lists
- Dictionaries
- ~*Tables*~

Collect data in a structured way (if possible)

Collecting
sign-in sheets, forms + surveys, toggl.com, spreadsheets

Analyzing
SQL, R (+ the tidyverse), python (+ the pandas package), the shell, and more...
4. Better Together
Talking about how peer mentoring & instruction should happen. #collabw16

Like a pyramid scheme, but for good rather than evil.

https://twitter.com/swcarpentry/status/711957318724919296
Three proverbs

If it ain’t broke, don’t fix it. If it’s already written, don’t write it yourself.
  ● Use pre-existing tools (software, libraries, expertise, training, etc.)

Two heads are better than one.
  ● Use other people’s ideas

“Though one may be overpowered, two can defend themselves. A cord of three strands is not quickly broken.” Ecclesiastes 4:12
  ● There is power in forming coalitions and communities.
Put people at the center

From “Best Practices for Scientific Computing”

Write Programs for People, Not Computers

Scientists writing software need to write code that both executes correctly and can be easily read and understood by other programmers (especially the author's future self).

From “Research Computing Facilitators: The Missing Human Link in Needs-Based Research Cyberinfrastructure”

...facilitators must be needs-focused and not CI [Cyberinfrastructure] solutions-focused [...] Only after a ‘diagnosis’ of needs for CI capabilities has been performed should CI resources be considered [...] Recommendations should be made with a focus on and communication of the greatest potential for transformative impact to the scholar's work.
In summary

1. Organize and document your work
2. Automate things
3. Structure your data
4. Prioritize people
Final questions

What processes do you use in your work?

Have you ever thought about them as generalizable “best practices”?

Did any of the things I just talked about resonate with you?

Is there something you want to try?

How can you keep thinking about these ideas?
Thanks for Listening

What questions do you have?
Resources

- **Software Carpentry**
  - [https://software-carpentry.org/](https://software-carpentry.org/)
  - [https://software-carpentry.org/lessons](https://software-carpentry.org/lessons)

- **Data Carpentry**
  - [http://www.datacarpentry.org/](http://www.datacarpentry.org/)
  - [http://www.datacarpentry.org/lessons](http://www.datacarpentry.org/lessons)

- **“Best Practices in Scientific Computing”**
  - [http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1001745](http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1001745)

- **“Good Enough Practices in Scientific Computing”**

- **Reproducible Research**
  - [http://kbroman.org/steps2rr](http://kbroman.org/steps2rr)

- **Organizing + Naming Things:**
  - [https://github.com/jennybc/organization-and-naming](https://github.com/jennybc/organization-and-naming)

- **Software and Data Carpentry at UW Madison**
  - [https://aci.wisc.edu/data-software-carpentry-workshops/](https://aci.wisc.edu/data-software-carpentry-workshops/)

- **“Research Computing Facilitators: the Missing Link in Needs-Based Research Cyberinfrastructure”**

Contact Christina at ckoch5 (at) wisc (dot) edu