Better Software, Better Data Handling

Slides DOI: 10.5281/zenodo.4282599

20 November 2020, CODATA - Webinar Series: Research Skills
(https://codata.org/initiatives/strategic-programme/codata-connect/webinar-series-research-skills/)

Shoaib Sufi, UK Software Sustainability Institute
ORCID: 0000-0001-6390-2616 | shoaib.sufi@software.ac.uk
The Software Sustainability Institute

A national facility for cultivating world-class research through software

• “Better Software, Better Research”
• Software code/processes/community reaches boundaries in its development that prevent improvement, growth and adoption
• Providing the expertise and services needed to negotiate to the next stage
• Programmes, events, policy, guidance and tools to support the community developing and using research software
• We advocate for all things Research Software

Supported by the UK Research Councils through grants EP/H043160/1, EP/N006410/1 and EP/S021779/1, with additional project funding from Jisc. A collaboration between the universities of Edinburgh, Manchester, Oxford and Southampton.
<table>
<thead>
<tr>
<th>Teams</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Software</strong></td>
<td><strong>Software</strong></td>
</tr>
<tr>
<td>Helping the community to develop software that meets the needs of reliable, reproducible, and reusable research</td>
<td>75+ project consultancies</td>
</tr>
<tr>
<td><strong>Policy</strong></td>
<td><strong>Policy</strong></td>
</tr>
<tr>
<td>Collecting evidence on and promoting the place of software in research &amp; sharing with stakeholders</td>
<td>1500+ RSEs engaged</td>
</tr>
<tr>
<td><strong>Outreach</strong></td>
<td><strong>Outreach</strong></td>
</tr>
<tr>
<td>Exploiting our platform to enable engagement, delivery &amp; uptake</td>
<td>170+ external contributors</td>
</tr>
<tr>
<td><strong>Training</strong></td>
<td><strong>Training</strong></td>
</tr>
<tr>
<td>Delivering essential software skills to researchers, partnering with institutions, doctoral schools and the community</td>
<td>300+ Carpentry workshops</td>
</tr>
<tr>
<td><strong>Community</strong></td>
<td><strong>Community</strong></td>
</tr>
<tr>
<td>Developing Communities of Practice by supporting the right people to understand and address topical issues</td>
<td>140+ Fellows</td>
</tr>
</tbody>
</table>

The "7/10"

Would be impossible, 68%
No effect, 11%
Possible, but difficult, 21%
Better Software,
Better Data Handling
Today’s Journey

• Spreadsheets
• Other options
• Resources and Training
• Data Carpentry
• Pedagogy practice and training
• Other initiatives

Photo by Zbysiu Rodak on Unsplash
Spreadsheets
Spreadsheets - data problems

• Microsoft Excel autocorrecting gene names to dates!
  ▪ Like MARCH1 — short for “Membrane Associated Ring-CH-Type Finger 1” — Excel converts that into a date: 1-Mar
  ▪ One study from 2016 examined genetic data shared alongside 3,597 published papers and found that roughly one-fifth had been affected by Excel errors!
Spreadsheets - format problems

• Microsoft Excel file format caused 16,000 Covid19 cases in the UK to be lost
  ▪ Use of XLS (65K rows) vs XLSX (1M+ rows) for integrating results
  ▪ limit reached - rows just discarded

• Delayed contact tracers knowing who to contact

www.bbc.co.uk/news/technology-54423988

Excel: Why using Microsoft's tool caused Covid-19 results to be lost

By Leo Kelion
Technology desk editor
5 October

Coronavirus pandemic

Software Sustainability Institute
Spreadsheets can be used properly

- Courses & books are available
- But the majority of people do not use best practices in spreadsheets, probably because it so easy not to!
- Spreadsheets can be done in so many different ways!
Better options
Better tools & languages

- A Scripted approach
  - Reproducible
  - Easier to compare versions
  - A more consistent version for sharing

- The R Project for Statistical Computing
- Python
- Mathworks Matlab
- GNU Octave
Resources and Training
So you want to learn

Places to look: (that you can fit in with your day job!)

- Courses by local University IT department for ECR’s
- Research Community based learning initiatives
- Self directed Learning

Out of scope for this talk:

- Fully fledged courses (that take up 30-100% of your time for more than a month) ← day job?
Research led training communities

- The Carpentries
  - Software
  - Data
  - Library
- Code Refinery
- Our Code Club

Software Sustainability Institute
Online course review sites

- Online review sites
  - Course talk
  - Class Central
  - Recommends and Rankings help choose
- MOOCs & more
  - Coursera
  - EdX
  - Future Learn etc

Software Sustainability Institute
Autodidactic

- Autodidactic
  - self taught - usually complex topics e.g. calculus or a language.
  - 15%? 70%?
- The need for training & community
  - Get feedback
  - Clear blockages in your understanding
  - Builds confidence
  - Help form Learning communities

Experiences with efficient methodologies for teaching computer programming to geoscientists

Christian T. Jacobs, Gerard J. Gorman, Huw E. Ross, Lorraine Craig

Affiliation for authors #1, #2 and #4: Department of Earth Science and Engineering, South Kensington Campus, Imperial College London, London, SW7 2AZ

Affiliation for author #3: Educational Development Unit, South Kensington Campus, Imperial College London, London, SW7 2AZ

Corresponding author email address: c.jacobs01@imperial.ac.uk

Short Title: Efficient methodologies for teaching programming to geoscientists

Paper type: Curriculum & Instruction (Instructional Approaches)

Keywords: computer programming, undergraduate, teaching methodology, feedback

Manuscript accepted for publication

in the Journal of Geoscience Education on 9 June 2016

Abstract

Computer programming was once thought of as a skill required only by professional software developers. But today, given the ubiquitous nature of computation and data science it is quickly becoming necessary for all scientists and engineers to have at least a basic knowledge of how to program. Teaching how to program, particularly to those students with little or no computing background, is well-known to be a difficult task. However, there is also a wealth of evidence-based teaching practices for teaching programming skills which can be applied to greatly improve learning outcomes and the student experience. Adopting these practices naturally gives rise to greater learning efficiency - this is critical if programming is to be integrated into an already busy geoscience curriculum. This paper considers an undergraduate computer programming course, run during the last 5 years in the Department of Earth Science and Engineering at Imperial College London. The teaching methodologies that were used each year are discussed alongside the challenges that were encountered, and how the methodologies affected student performance. Anonymised

https://arxiv.org/abs/1505.05425 -
The Carpentries approach

- Instruction
- Material for reference
- Learn by doing
- Helpers to clear up understanding

Evidence of Carpentries' Impact on Learners

- 66.2% of respondents use programming languages and/or the command line to automate repetitive tasks.
- 49.3% of respondents have improved their data management and project organisation.
- 46.1% of respondents use version control to manage code.

Pre and Post Comparison of Skills and Perception

carpentries.org/blog/2018/07/evidence-impact
Data Carpentry
Data Carpentry (DC)

- Different Curriculums
  - Mature - ‘2’ days
    - Ecology, Genomics, Social Sciences, Geospatial
  - In development - ‘2’ days’
    - Image processing, Economics, Astronomy, Digital Humanities and more
  - Semester long
    - Biology

All About Data Literacy!
# A typical DC workshop

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Site</th>
<th>Repository</th>
<th>Reference</th>
<th>Instructor Notes</th>
<th>Maintainer(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ecology Workshop Overview</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Karen Cranston, Aleksandra Pawlik, Tracy Teal, Ethan White, Fabrice Rwasimitana</td>
</tr>
<tr>
<td>Data Organization in Spreadsheets for Ecologists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Christie Bahlai, Peter R. Hoyt, Tracy Teal</td>
</tr>
<tr>
<td>Data Cleaning with OpenRefine for Ecologists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cam Macdonell, Deborah Paul, Phillip Doehle, Rachel Lombardi</td>
</tr>
<tr>
<td>Data Management with SQL for Ecologists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Donal Heidenblad, Timothée Poisot, Rémi Rampin, Christina Koch, Katy Felkner</td>
</tr>
<tr>
<td>Data Analysis and Visualization in R for Ecologists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ana Costa Conrado, Auriel Fournier, François Michonneau, Brian Seok</td>
</tr>
<tr>
<td>Data Analysis and Visualization in Python for Ecologists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Tania Allard, Maxim Belkin</td>
</tr>
</tbody>
</table>

Some material is available in Spanish also - and you tend to do R or Python - ideally 2.5 days for the workshop
Using Spreadsheets in Research

- Data organisation or Data ‘wrangling’
  - The ‘sweet spot’ for spreadsheets
- Data exported for Analysis elsewhere
  - Adaptation and reproducibility is hard
  - Easy to reference wrong cells in calculations
    - Much easier to pick up this type of error using a scripting approach (e.g R, Python)
- Data presentation
  - Not optimal, use document editor for presentation
- Using Spreadsheets for “quick and dirty” analysis is OK - don’t consider it final and good data organisation helps here!
**Good Data Organisation**

- Don’t modify RAW data directly
- Take a copy and make changes to that to make a ‘clean’ data set to analyse
- Keep track of changes between RAW and ‘clean’ by keeping notes in a text file recording the steps you took to move from RAW to ‘clean’

**Keep Data ‘Tidy’**
- Variable in columns
- Observation in each row
- Don’t combine data into one cell
- export the data to a text-based format e.g CSV

**General rules:**
- columns = variables
- rows = observations
- cells = data (aka values)

"It is often said that 80% of data analysis is spent on the process of cleaning and preparing the data" (Dasu and Johnson 2003).
Common Formatting Problems

• Good formatting makes cleaning & analysis easier
• Multiple small tables breaks the one row per observation rule
• Keep all observation in one tab for a particular experiment
  ▪ minimise joining
  ▪ maintains consistency
• Zero vs null
  ▪ and how to represent when you don’t capture values
• Formatting
  ▪ Using formatting to represent data ← fix: new column
  ▪ Merged cells ← fix: avoid
  ▪ Units in cells ← fix: same unit in the column or new unit column
  ▪ Avoid comments ← use a new column
More formatting

- Choose good column names
  - avoid spaces, make them meaningful, include units if possible,
    use a naming convention
- Copy and paste
  - remove formatting - use a cell as a holder of text and spaces
- Other files
  - Data files
  - Metadata files ← column name meanings, unit, exceptions, etc
  - A readme.txt to explain what each file contains and any relationships
- Date format
  - Use different columns: data, month, year or year and day of year
Better Data

- Data validation
  - restrict the options or range
- Quality control
  - Remember to do this in a different file
  - Document your steps
- Sorting
  - Expand your sort ← maintain one row as one observation
  - Look at the start and end ← where errors tend to hide
- Conditional formatting

Guide to writing "readme" style metadata

A readme file provides information about a data file and is intended to help ensure that the data can be correctly interpreted, by yourself at a later date or by others when sharing or publishing data. Standards-based metadata is generally preferable, but where no appropriate standard exists, for internal use, writing "readme" style metadata is an appropriate strategy.

Want a template? Download one and adapt it for your own data!

- Best practices
- Recommended content
  - General information
  - Data and file overview
  - Sharing and access information
  - Methodological information
  - Data-specific information
- References
- Related information

data.research.cornell.edu/content/readme
Exporting data

• For analysis in other programs
  ▪ universal, open, static format
  ▪ Comma Separated Values - CSV or Tab Separated Values - TSV is a good choice
  ▪ You can open them in e.g. Excel again - but remember any changes won’t be saved.
  ▪ Be careful about line endings in CSV files
    • LF (Unix) vs CR LF (Windows)
OpenRefine - cleaning messy data

- Semi-automated cleaning that saves time
- Cleans
- Formats
- Tracks changes
- Does not overwrite raw data

Key features:
- Dataset overview
- Resolve inconsistencies
- Split data into more granular parts
- Match local data to other sets
- Enhance data from other sources
- Automation-replay steps on multiple files

"Many people comment that this tool saves them literally months of work trying to make these edits by hand."
Two main DC lessons around analysis

- **Python**
  - General purpose language with data analysis libraries
  - Great libraries and editors - e.g. JupyterLab, Spyder, Visual Studio Code

- **R**
  - Built as a statistical computing language can be a bit strange to do general purpose things in
  - Great libraries and editors - R Studio

Software Sustainability Institute
Data Analysis and Visualization in Python

- Python Syntax
- Jupyter notebook interface
- Importing CSV files
- **pandas** library to work with data frames
- Summary info from data frames
- An intro to plotting

See: [datacarpentry.org/python-ecology-lesson/index.html](http://datacarpentry.org/python-ecology-lesson/index.html)
Other tools and approaches

- Further DC:
  - SQL ← a different approach to querying data
  - R ← similar place to Python in Analysis

Better software skills also help - more in the region of Software Carpentry -
- The Unix Shell ← automation
- Git ← version control
- Python / R ← more of a programming focus
- Reproducibility in R

Core

Lesson

- The Unix Shell
- Version Control with Git
- Programming with Python
- Plotting and Programming in Python
- Programming with R
- R for Reproducible Scientific Analysis

Additional

- Automation and Make
- Programming with MATLAB
- Using Databases and SQL

Software Sustainability Institute

www.software.ac.uk

Other tools and approaches

- Further DC:
  - SQL ← a different approach to querying data
  - R ← similar place to Python in Analysis

Better software skills also help - more in the region of Software Carpentry -
- The Unix Shell ← automation
- Git ← version control
- Python / R ← more of a programming focus
- Reproducibility in R

Core

Lesson

- The Unix Shell
- Version Control with Git
- Programming with Python
- Plotting and Programming in Python
- Programming with R
- R for Reproducible Scientific Analysis

Additional

- Automation and Make
- Programming with MATLAB
- Using Databases and SQL

Software Sustainability Institute

www.software.ac.uk
Pedagody practice and training
Beyond learning

- Teaching training and experience - help transition from postdoc to faculty
- CV worthy material

Commitment

- Attendee
- Helper
- Organiser
- Instructor
- Curriculum developer
- Instructor trainer
- Exec Committee

Complexity

Software Sustainability Institute
Teaching Infrastructure

carpentries.org/become-instructor

carpentries.github.io/instructor-training

- Introduce you to evidence-based best-practices of teaching.
- Teach you how to create a positive environment for learners at your workshops.
- Provide opportunities for you to practice and build your teaching skills.
- Help you become integrated into the Carpentries community.
- Prepare you to use these teaching skills in teaching Carpentries workshops.

carpetes.org/community-lessons

cdh.carpentries.org

developers.carpentries.org

github.com/carpentries/styles

The Carpentries Incubator

Development guidebook

Software Sustainability Institute
Community Discussions

carpentries.org/community_discussions

1. **Pre- and Post-Workshop Discussions** These discussions are designed for those getting ready to teach or having recently taught to come discuss their workshop with the community. They occur twice per week.

2. **Themed Discussion Sessions** These discussions are centered around a particular topic ranging anywhere from teaching your first workshop to community building strategies. They occur once per month.

3. **Carpentries Conversations** These Conversations are hosted by one of our Committees or Task Forces to provide the community with the opportunity to learn about and discuss new developments and programs in our organisation. They occur once per month.

twitter.com/thecarpentries

swc-slack-invite.herokuapp.com

Software Sustainability Institute
Other initiatives
Open Science & Reproducibility

Open Science / Research
- Open Access
- Open Data
- Open notebook science
- Open Source
- It’s about transparency and access

Benefits:
- Verification
- Reduce duplication
- Reuse
- Trustworthiness
- Quality

International Level

National & Institutional

Problems:
- Publication Bias
- Low statistical power
- P-value hacking
- Harking (hypothesis after results are known)

Institutional & Grassroots

- Training
- Best practice / primers
- Culture
- Researcher led
  - Local network model

started 2018, 109 institutions in 25 different countries

www.oecd.org/science/inno/open-science.htm
www.ukrn.org
reproducibilitea.org

Software Sustainability Institute
FAIR - Findable, Accessible, Interoperable, Reusable

FAIR (2015)

Turning FAIR into reality (2018)

FAIR 4 Research Software (2019)

FAIR for Research Software (FAIR4RS) WG

www.rd-alliance.org/groups/fair-research-software-fair4rs-wg

3 subgroups:
- How do FAIR principles map to Software
- How has FAIR been applied to workflows, notebooks, training etc
- Definition of research software

Why is this important?:
- Understanding how to make your analysis FAIR will help make it Reproducible and mindfully Open

www.nature.com/articles/sdata201618

op.europa.eu/s/oriv

Software Sustainability Institute
In conclusion

• Better ways to handle and analyse data
• Learn best practices
• Make your work reproducible
• Get involved in training communities for career credit
• Be aware of the wider context
• Do what you do better - make coding/scripting/ aka better software your data handling superpower!

Photo by Miguel Bruna on Unsplash

Software Sustainability Institute
Acknowledgements

The SSI team/alumni:
- Agata Dybisz
- Aleksandra Nenadic
- Aleksandra Pawlik
- Alexander Hay
- Ania Brown
- Arno Proeme
- Carole Goble
- Caroline Jay
- Claire Wyatt
- Clem Hadfield
- Dave De Roure
- Devasena Prasad
- Giacomo Peru
- Graeme Smith
- Iain Emsley
- Jacalyn Laird
- James Graham
- John Robinson
- Les Carr
- Lucia Michielin
- Malcolm Atkinson
- Malcolm Illingworth
- Mario Antonioletti
- Mark Parsons
- Mike Jackson
- Olivier Philippe
- Priyanka Singh
- Rachael Ainsworth
- Raniere Silva
- Rob Baxter
- Robin Wilson
- Sam Manghan
- Selina Aragon
- Shoaib Sufi
- Simon Hettrick
- Stephen Crouch
- Tim Parkinson
- Toni Collis
- Plus the SSI Fellows
- and RSE community

Supported by the UK Research Councils through grants EP/H043160/1, EP/N006410/1 and EP/S021779/1.

Additional project funding received from Jisc.
Questions?