Automation

and

Reproducibility

We sometimes optimize the wrong things

In the Seven Years' War, 1754-1763...

Britain lost 1,512 sailors to enemy attacks.

We sometimes optimize the wrong things

In the Seven Years' War, 1754-1763...

Britain lost 1,512 sailors to enemy attacks.

*...and nearly **100,000** to scurvy!*

We sometimes optimize the wrong things

- What are some things we can optimize in software development?
  - Computing time (usually fairly cheap)
  - Programmer time (expensive)
  - Cognitive load
    - Your brain can juggle about 7 ± 2 chunks of information at once
  - Accessibility
- Whatever we choose to optimize, there should be a reason

Why automate?

- Optimize programmer time: let the machine handle things without your supervision
  - e.g. on a computing cluster
- Optimize cognitive load: record those pesky command line options that you can never seem to remember, and forget them!
- For yourself – *repeatability*
- For others – *reproducibility*

Reproducibility

“Commonly research involving scientific computations are reproducible in principle, but not in practice.”

“In our laboratory, we noticed that after a few months or years, researchers were usually unable to reproduce their own work without **considerable agony**.”

Schwab, Matthias, et al. "Making scientific computations reproducible." Computing in Science & Engineering 2.6 (2000): 61-67.

Real tools for reproducibility

- Shell scripting: “why not just do this from the file browser?”
  - Record a series of actions and you or someone else can repeat those actions later with precision
  - Command-line utilities provide a powerful way to manipulate and analyze files quickly

Real tools for reproducibility

- Version control (git): “why not just use (Dropbox/a USB drive/e-mail attachments)?”
  - Share your data and code with others, easily!
    - You can limit this to just collaborators or release it to the public
  - Provenance: keep a record of every change you make to a file, and *why* you made the change

Real tools for reproducibility

- Automated build system (Make): “why not just write an R script to build everything?”
  - Run and re-run only the parts of a data analysis pipeline that you need to
  - When you come back to your code in a week, a month, or a year, will you remember which script generates which data file?
  - A README is good; a single build command is even better

Be a skeptic

- We're claiming that we can improve your efficiency as researchers; do you believe us? Why or why not?
- Make us convince you – don't just take our word for it!
- Do these tools provide an improvement over what you use now?