Programming FUNdamentals

Dav Clark

Rochelle's Law (paraphrased)

The difference between "clueless users" and "tech support" is a willingness to google for a few minutes longer.

D-Lab Offerings

- We are a “front desk” for social scientists
- Training opportunities:
  - Workshops and Intensives (like this)
  - Working groups
  - Consulting
- A space for you & your team

Let’s get a sense of where you’re at...

Topics - Stuff on your computer

- GUI? Huh?
- Files, folders and directories - organized so you can find them again!
- What is a terminal / command prompt?
- What's a text editor / IDE / "application"?
- Pros and cons of using applications vs. using programming / scripting.

Topics - Data & Networking

- Overview of storage formats: XML, plain text, "document formats," CSV, binary formats (e.g., HDF5) and databases.
- How big is my data, really? And how does this relate to networking, storage, and compute choices?
- How does the web work? What is a server?

Files and folders (names for things)

- We live in a world of search, and that often works...
- But it’s better to have two ways to find things in case one doesn’t work!
- Usually it won’t take much time to give something a long, descriptive name
- What do you do that works?

GUI Interfaces

- Graphical User Interface
- In short
  - Graphical = visual, usually 2D
  - User Interface = A way for *you* to interact with the computer
- What's the alternative?
- Text!

Terminal / Command Prompt

- Who here has a command prompt program installed?
  - OS X: “Terminal.app”
  - Windows: “powershell.exe” (and “cmd.exe”), but **Git Bash** is better!
  - Other systems: something with “term” in it
- Basic commands: `ls`, `pwd`, **`man <cmd>`** **(‘q’ to exit)**, `cd`, `rm`, `mv`, `cp`
- These usually correspond to things you can do graphically

GUIs	Textual Commands
Easier to “discover”	No need to remember where anything is - just type it
Hard to use over network (except via web)	Easy to use over ssh (a networked command prompt)
Mistakes are often more obvious	Data often not visible

More on Text

Editing “text files”

- Who has a text editor installed?
  - Windows: “notepad.exe”
  - OS X: “TextEdit.app”
  - Other systems: probably “nano”
- But all of those are kinda lame...

1. Use whatever your colleagues use
2. Use Sublime Text 2 (and buy it)
3. But you’d rather have something “free?”
   a. Windows: Notepad++ (Atom soon)
   b. OS X: TextWrangler, TextMate 2, or Atom
   c. Linux, etc.: gedit, kate on KDE (Atom soon)

But besides text editors...

- GUIs for many programs (like SPSS, Stata) may edit / produce text files
- Other applications (like Excel) have the option to save “plain text” (like CSV)
- Special-purpose code editors (IDEs) for specific programming tasks
  - OS X: Xcode
  - Windows: Visual Studio
  - Java (and friends): Eclipse, IntelliJ

The benefits of text (for data)

1. Almost every environment can understand text
   a. Even if that's not the "default" format (cf. Office)
   b. BUT - encoding introduces some gotchas
2. Your computer comes with a text editor
3. Command line interfaces are text-based
4. Usually very obvious what's going on!

The disadvantages of text (for data)

1. It’s slow
2. Encoding issues

Binary

How Computers Really Think

You think	Computer thinks
a	01100001
97	01100001
“1,204.89”	00110001 00101100 00110010 00110000 00110100 00101110 00111000 00111001
	That same space can hold 16 digits * 10^{384}!

The benefits of binary data

1. It's how computers think:
   a. Faster
   b. More compact (smaller files for the same data)
2. Can have explicit metadata, avoiding encoding issues

Disadvantages of binary data

1. Often need special programs to read
2. If a file gets corrupted, you may lose **all** your data
3. It's generally not worth using until you have a really big speed or space problem!

Discuss file formats

Your pick:

- Binary
  - HDF5 (& NetCDF4)
  - Databases (not quite a file format)
  - Documents: doc, exc, rtf, etc.
- “Plain Text”
  - XML, HTML, etc.
  - JSON, YAML
  - Doc-like: CSV, txt (+ markdown, etc.)
  - Most programming languages

So, how big *is* your data?

Sadly, you may have to do some arithmetic…

- Conceptually map the amount of data you expect (e.g., number of observations * number of bytes per observation)
- Establish data transfer / processing rates using smaller samples
- Be careful of nonlinear demands
  - e.g., matrix math may require 4 (or more) times more processing for twice the data

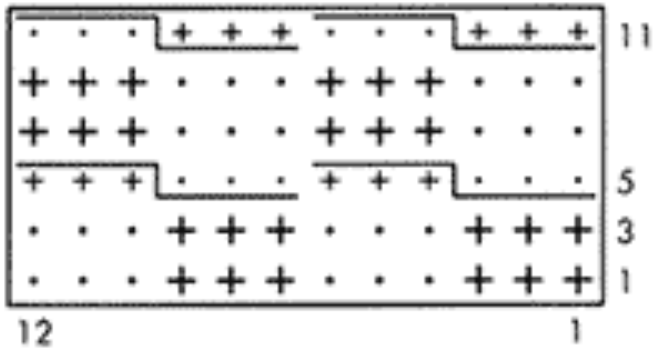
Programming

- Like a playbook or recipe
  - Multiple steps get done reliably
  - Almost always text based
- Common elements:
  - Names for things
  - Data
  - Containers for related data
  - Ways to efficiently repeat things - functions and loops
  - Ways to make decisions (if / then)
- Many ways for things to break / be confusing
- Start simple!

Common Knitting Abbreviations

bo - bind off
co - cast on
dec - decrease
k - knit
inc - increase
M1 - make 1
p - purl
PU - pick up
RS - right side
tog - together
WS - wrong side
yo - yarn over

=?


11
5
3
1
12
1

Dav's Law

If you already know how to get your work done, and can share that method with others transparently…

Just do it that way!

Error Messages (and what to do)

- google: name-of-program + text in error message
  - Remove user- and data-specific information first!
- See if you can find examples that do and don’t produce the error
- Stack Overflow (and friends)

A bit on networking and servers

- Everything you can “do” on a computer is “done” by one or more programs.
- Programs can run on your computer or on another computer you have access to:
  - E.g., web servers
- Some programs *even if they are running on your computer* like to “pretend” they are on a network (and actually, they are):
  - E.g., web servers
  - **Databases**