Library Carpentry
Introduction to Data

Where to go for help

Stickers
Helpers
Sticky notes
github.com/???

Week 1: Introduction to Data

18:00-18:45 Jargon Busting
18:45-19:30 Foundations
19:30-20:15 Regular Expressions

Jargon Busting

Teams of 5 or 6
Write terms you want busting on stickies
Cluster (retaining duplicates)
Discuss and explain
Note resolved terms
Note unresolved terms
Report back

Foundations

The Computer is Stupid
Why automate
Keyboard shortcuts are your friend
Plain text formats are your friend
Structuring files and folders

Foundations

The Computer is Stupid

ERROR

Foundations

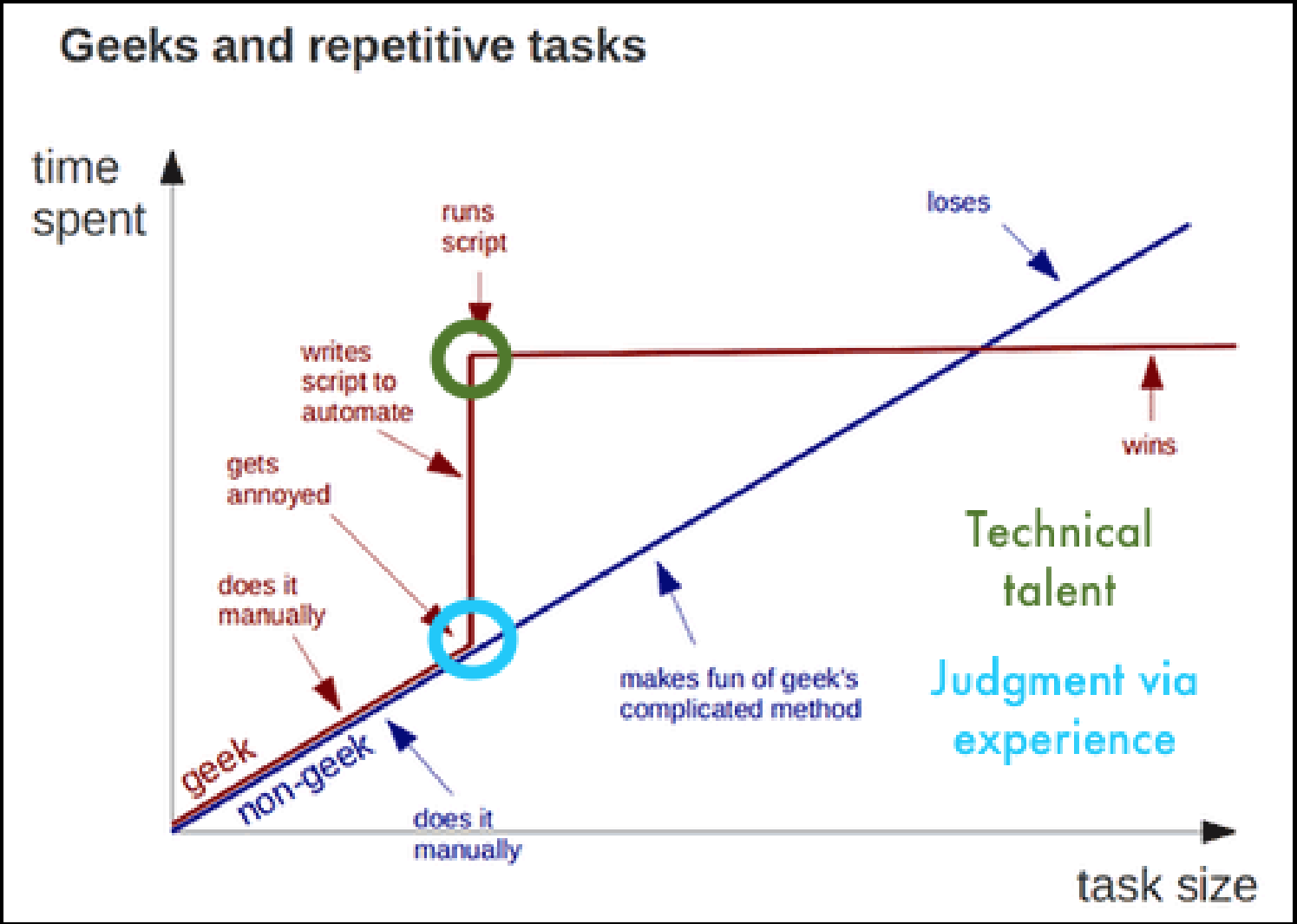
Why automate?

Borrow, borrow, borrow
There is no correct language
Professional development
Knowing some code ~ evaluating software
Making time to do fun stuff!

Andromeda Yelton, "Coding for Librarians: Learning by Example", *Library Technology Reports* 51:3 (April 2015), doi: 10.5860/ltr.51n3

Foundations

Why automate?



Credit: Andy Kirk

Foundations

Keyboard shortcuts are your friend

Efficiency and control

Foundations

Plain text formats are your friend

Computers process them better
Platform agnostic
Display orientated files aren't your friend
Markdown

Foundations

Structuring files and folders

Consistent and predictable data structure
Semantic-data hybrid directory names
Your own system is fine
Links files and directories with names

You are the most likely person to forget what you once did!

Regular Expressions

Match on types of character

Match patterns

Capture the parts that match your pattern

Regular Expressions

organi[sz]e

organise (match)
organize (match)
reorganise (match part so will also find)
reorganize (match part so will also find)

Regular Expressions

[ABC]	matches A or B or C.
[A-Z]	matches any upper case letter.
[A-Za-z0-9]	matches any upper or lower case letter or any digit.

Regular Expressions

.	matches any character at all.
\d	matches any single digit.
\w	matches any part of word character.
\s	matches any space, tab, or newline.
\b	matches a word boundary.
^	asserts start of the line.
$	asserts end of the line

Regular Expressions

^[Oo]rgani.e\b

Regular Expressions

* matches proceeding character any number of times including zero.

\+ matches proceeding character any number of times excluding zero.

? matches the proceeding character one or zero times.

{VALUE,VALUE} matches proceeding character a defined number of times.

| simply means or.

Regular Expressions

^[Oo]rgani.e\w*

Regular Expressions

[Oo]rgani.e\w+$

Regular Expressions

^[Oo]rgani.e\w?\b

Regular Expressions

^[Oo]rgani.e\w?$

Regular Expressions

\b[Oo]rgani.e\w{2}\b

Regular Expressions

\b[Oo]rgani.e\b|\b[Oo]rgani.e\w{1}\b

Regular Expressions

Teams of 4
Work through excersse
Split into two teams and write:

- strings that need regex
- regex that need outputs

Test each other!

After the session: take the quiz!