Find a Way to Develop Data Literacy - Using Library Carpentry!

ALC 2019

Abigail Sparling *MI*
Serials Metadata Librarian
University of Alberta

@abbisparling

Céline Gareau-Brennan *MLIS, MBA*
Public Services Librarian,
University of Alberta

@Celine_GB

UNIVERSITY OF ALBERTA

Learning Outcomes

1. Understand the value of Library Carpentry for improving data literacy

2. Learn how to use Library Carpentry for professional development

3. Explore tangible pedagogical techniques through a workshop demo

Session Plan

1. Introduction to Library Carpentry
2. Overview of Library Carpentry Pedagogies
3. Break (15 mins)
4. Workshop demo
    a. Data Jargon Busting
    b. Data Foundations
    c. Break (15 mins)
    d. Regular Expressions
5. Conclusion

Introduction to Library Carpentry

UNIVERSITY OF ALBERTA

What is Library Carpentry?

"Library Carpentry develops lessons and teaches workshops for and with people working in library- and information-related roles."

"Our goal is to create an on-ramp to empower this community to use software and data in their own work as well as be advocates for and train others in efficient, effective and reproducible data and software practices"

https://librarycarpentry.org/

UNIVERSITY OF ALBERTA

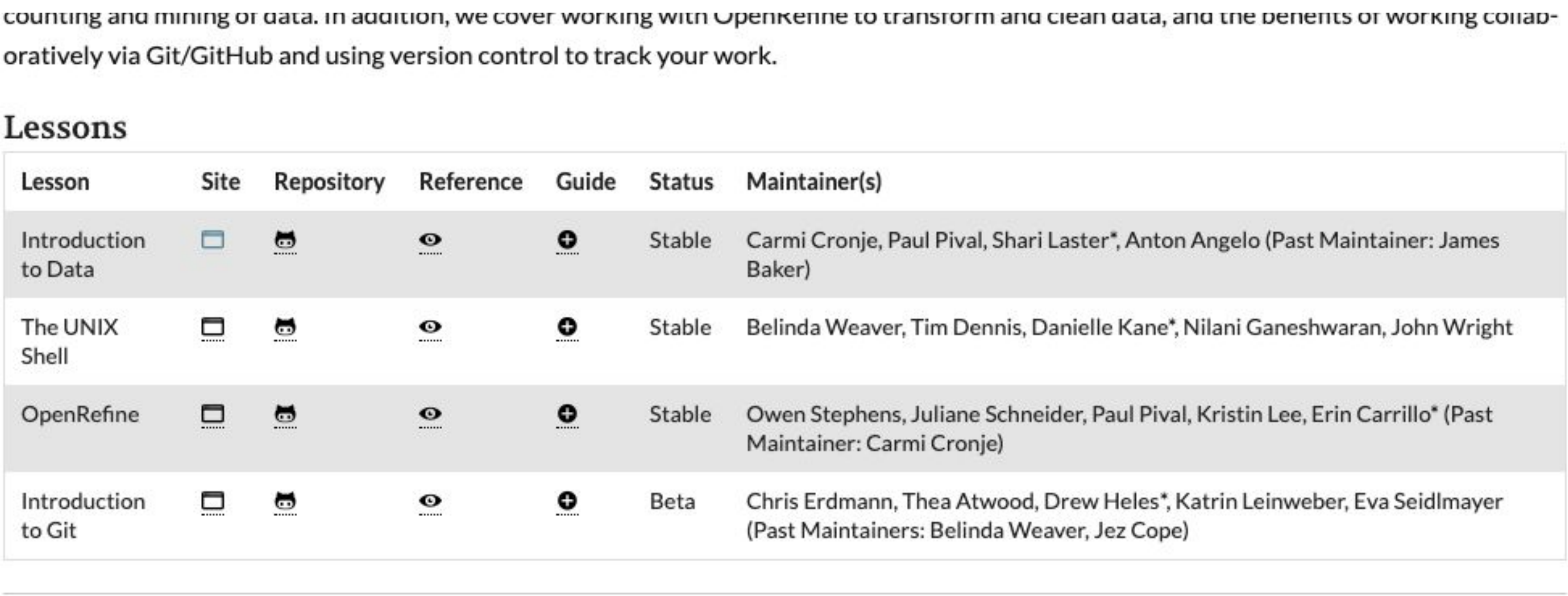
What do library staff gain from code?

UNIVERSITY OF ALBERTA

Resources at your disposal

1. Library Carpentry https://librarycarpentry.org/
2. The Carpentries https://carpentries.org/
3. The Carpentries Handbook https://docs.carpentries.org/index.html
4. The Carpentries Instructor Training https://carpentries.github.io/instructor-training/
5. American Library Association Library Technology Report called "Coding for Librarians: Learning by Example" https://thatandromeda.github.io/ltr/
6. Specifically for this Regular Expressions: Lynda.com - Course: Learning Regular Expressions https://www.lynda.com/Regular-Expressions-tutorials/Using-Regular-Expressions/85870-2.html?org=epl.ca

Resources at your disposal



counting and mining of data. In addition, we cover working with OpenRefine to transform and clean data, and the benefits of working collaboratively via Git/GitHub and using version control to track your work.

Lessons

Lesson	Site	Repository	Reference	Guide	Status	Maintainer(s)
Introduction to Data	▭	🐱	👁	⊕	Stable	Carmi Cronje, Paul Pival, Shari Laster*, Anton Angelo (Past Maintainer: James Baker)
The UNIX Shell	▭	🐱	👁	⊕	Stable	Belinda Weaver, Tim Dennis, Danielle Kane*, Nilani Ganeshwaran, John Wright
OpenRefine	▭	🐱	👁	⊕	Stable	Owen Stephens, Juliane Schneider, Paul Pival, Kristin Lee, Erin Carrillo* (Past Maintainer: Carmi Cronje)
Introduction to Git	▭	🐱	👁	⊕	Beta	Chris Erdmann, Thea Atwood, Drew Heles*, Katrin Leinweber, Eva Seidlmayer (Past Maintainers: Belinda Weaver, Jez Cope)

Overview of Library Carpentry Pedagogies

UNIVERSITY OF ALBERTA

The Carpentries Pedagogical Model

1. Applied approach
2. Learners are able to practice what they are learning in real time
3. Learners help each other during the workshop
4. Incorporates assessments within the lesson materials
5. Flexible & Responsive

Types of Learners

Novice

“I'm not sure what questions to ask”

Competent Practitioner

“I'm pretty confident, but I still look stuff up a lot!”

Expert

“I've been doing this on a daily basis for years!”

Experience level

Image credit: https://carpentries.github.io/instructor-training/02-practice-learning/index.html

What are the Pros & Cons (as a learner or teacher) of being:

1) a novice?
2) a competent practitioner?
3) an expert?

Mental Models of Learners

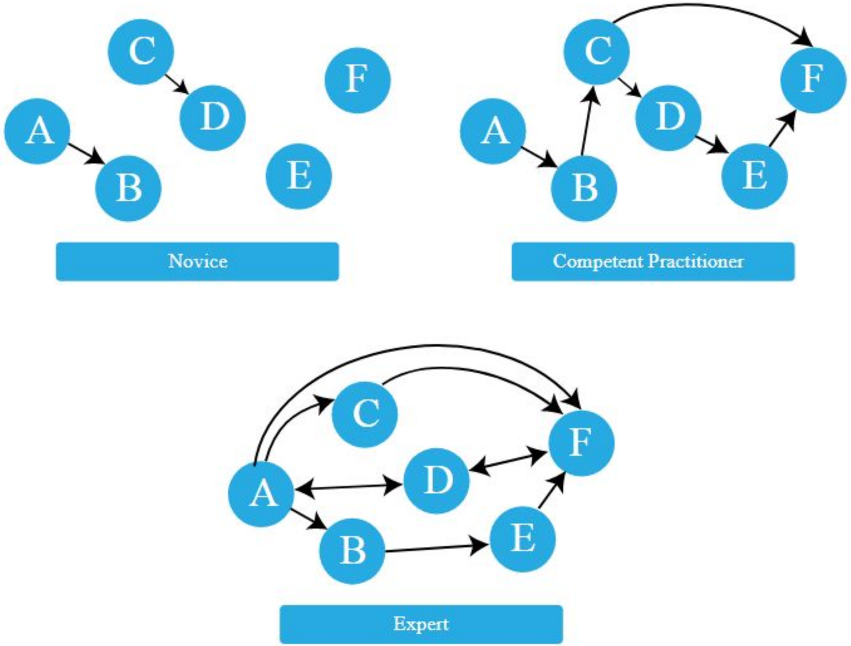


Image credit: https://carpentries.github.io/instructor-training/02-practice-learning/index.html

How “Knowledge” Gets in the Way

Types of misconceptions:

1. Simple factual errors

   Easiest to correct.

2. Broken models

   Address these by having learners reason through examples to see contradictions.

3. Fundamental beliefs

   Hardest to change.

Formative Assessments

1. Happens during teaching and learning.
2. Provide guidance to the instructor and the learner about what to focus on
3. Happens frequently
4. Takes different forms

What should we do as instructors if the class chooses:

1. mostly one of the wrong answers?
2. mostly the right answer?
3. an even spread among options?

Memory and Cognitive Load

1. Most of us store only a few items in short-term memory for a few seconds before we lose them.
2. Things seen together are remembered in chunks.
3. Teaching consists of loading short-term memory and reinforcing it long enough for items to be transferred to long-term memory.
4. Use Guided Practice to reduce cognitive load
5. Use formative assessments to avoid overloading short-term memory.

Formative Assessment

1. What are the 3 types of learners?

2. In Library Carpentry, what type of misconception do we address?

3. What is Guided Practice?

Motivation and Demotivation

Creating A Positive Learning Environment

1. *Presenting the instructor as a learner*
2. *Establishing norms for interaction*
3. *Encouraging learners to learn from each other*
4. *Acknowledging when learners are confused*

Teach the Most Useful First

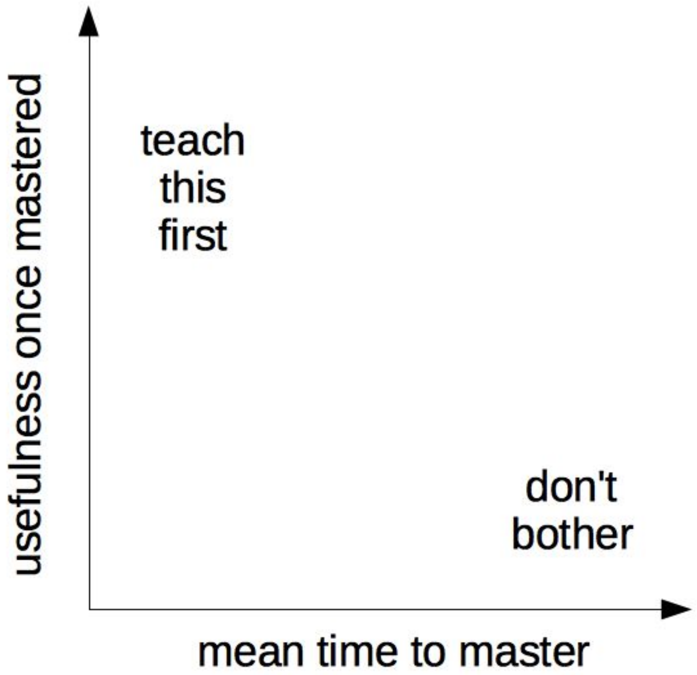


Image credit: https://carpentries.github.io/instructor-training/08-motivation/index.html

Brainstorming Motivational Strategies

Think back to a computational (or other) course you took in the past, and identify one thing the instructor did that motivated you.

Pair up with your neighbor and discuss what motivated you.

How Not to Demotivate Your Learners

- Talk contemptuously or with scorn about any tool
- Dive into complex or detailed technical discussion
- Pretend to know more than you do
- Use the J word ("just") or other demotivating words
- Take over the learner's keyboard
- Criticize students for making errors.

Psychological Demotivators

Stereotype threat - Reminding people of negative stereotypes, even in subtle ways, can make them anxious about the risk of confirming those stereotypes, in turn reducing their performance.

Imposter Syndrome - Imposter syndrome is the belief that one is not good enough for a job or position, and that one's achievements are due to luck rather than talent or skill.

Live Coding

WHAT: Working through the lesson material, typing in the code or instructions, with the workshop participants following along

WHY: Provides learners with **continuous opportunities for practice** (every time they type in a line of code or work through an exercise) and **continuous feedback** (their code either works or fails with an error message)

Tips for Successful Live Coding

- Go slowly
- Mirror your learner’s environment
- Use your screen wisely
- Stick to lesson material
- Leave no learner behind - use stickies
- Embrace mistakes
- Have fun!

Building Skills with Feedback

- Surveys
- Minute Cards / One-Up, One-Down
- Be explicit about discussing/responding to feedback

Give Us Feedback!

Write one thing you learned that you've found useful on your blue sticky note and one question you have about the material on the yellow sticky note

BREAK
15 minutes

https://bit.ly/2vrZsJf

UNIVERSITY OF ALBERTA

Workshop demo

UNIVERSITY OF ALBERTA

Goals

1. Defining terms, phrases, and concepts in software development and data science

2. Identifying and using best practices in data structures

3. Using regular expressions in searches

Jargon Busting

What terms, phrases, or ideas around code or software development have you come across and perhaps feel you should know better?

Key Takeaways

It helps to share what you know and don’t know about software development and data science jargon.

Data Foundations

UNIVERSITY OF ALBERTA

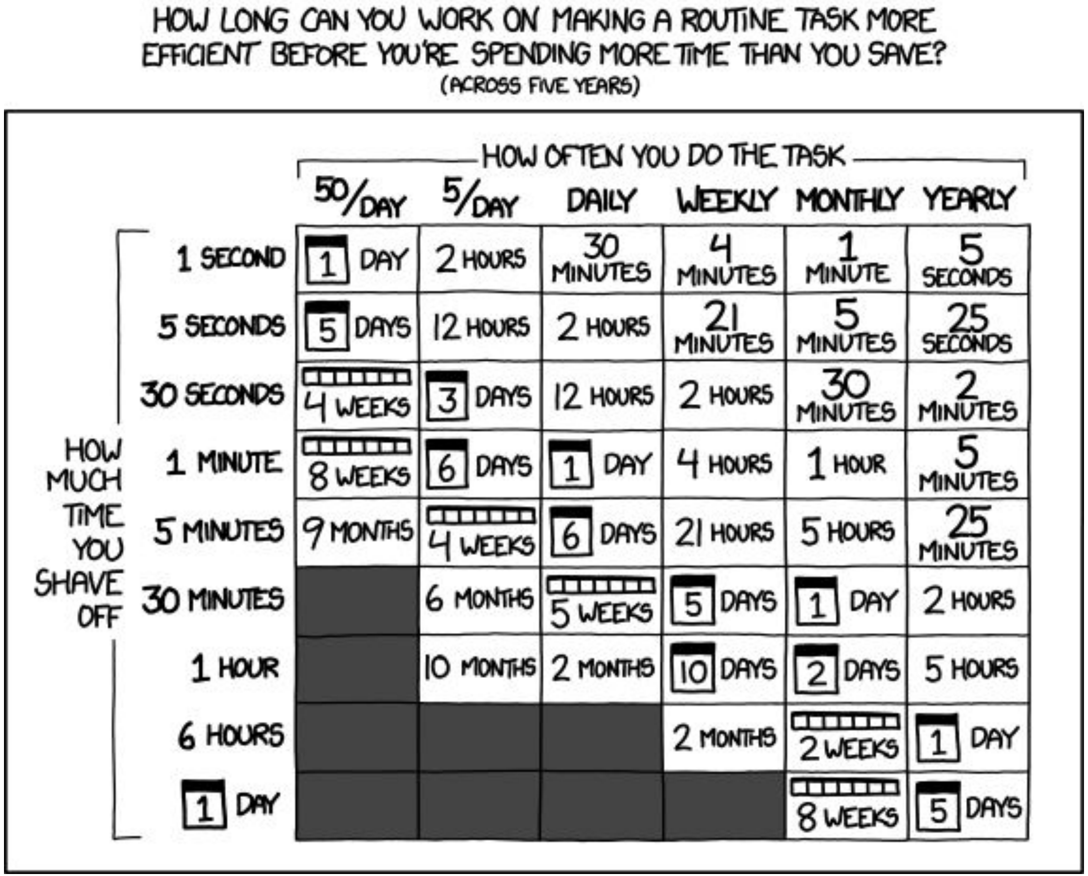
The computer is a machine.

People can interpret. Machines don’t.

When to take an automated or computational approach?

1. You know how to automate the task

2. You think this is a task you will do over and over again. AKA automating the task will save you time.



HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

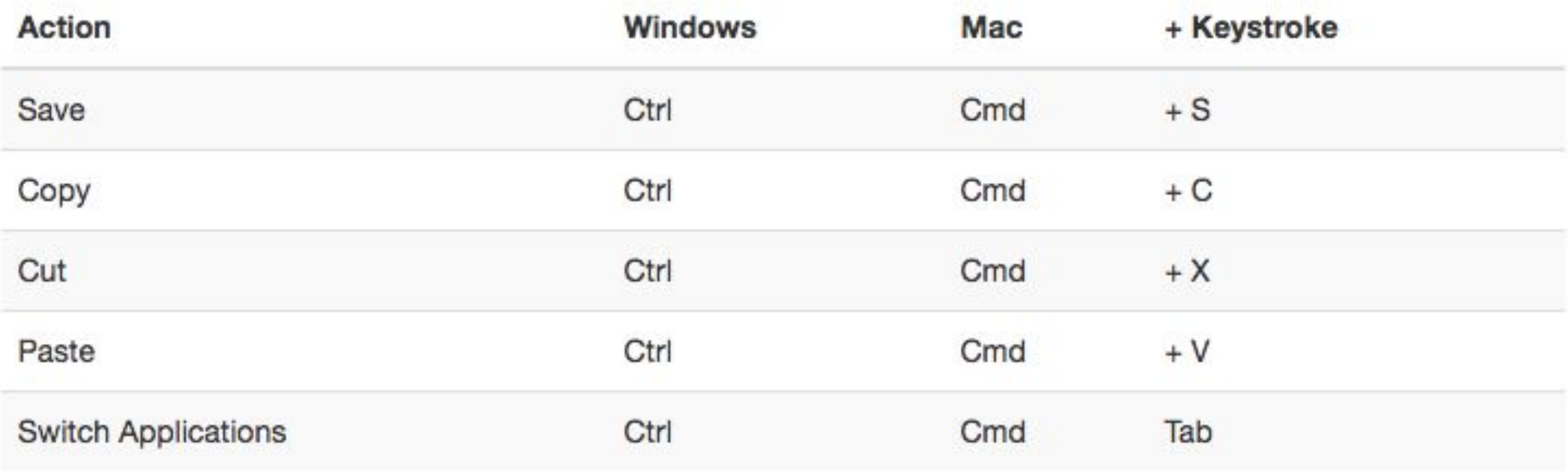
HOW MUCH TIME YOU SHAVE OFF	HOW OFTEN YOU DO THE TASK: 50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
30 MINUTES	■	6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
1 HOUR	■	10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
6 HOURS	■	■	■	2 MONTHS	2 WEEKS	1 DAY
1 DAY	■	■	■	■	8 WEEKS	5 DAYS

‘Is it worth the time?’ by Randall Munroe available at https://xkcd.com/1205/ under a Creative Commons Attribution-NonCommercial 2.5 License.

Main lessons for building data and software literacy

1. Borrow, borrow, and borrow again.
2. The correct language to learn is the one that works in your local context.
3. Consider the role of programming in professional development.
4. Knowing (even a little) code helps you evaluate projects that use code.
5. Automate to make the time to do something else!

Keyboard shortcuts are your friend



Action	Windows	Mac	+ Keystroke
Save	Ctrl	Cmd	+ S
Copy	Ctrl	Cmd	+ C
Cut	Ctrl	Cmd	+ X
Paste	Ctrl	Cmd	+ V
Switch Applications	Ctrl	Cmd	Tab

https://librarycarpentry.github.io/lc-data-intro/03-foundations/index.html

Plain text formats are your friend

Use platform-agnostic formats such as

- **.txt** for notes
- **.csv** (comma-separated values) or
  .tsv (tab-separated values) for tabular data.

Applications for writing and reading plain text files

1. For Windows users,
   a. Notepad++ http://notepad-plus-plus.org/

2. Mac or Unix users may find
   a. Komodo Edit
   b. Text Wrangler
   c. Kate, or Atom

Check-In

1. Under what conditions should we consider automating tasks?
2. What’s the best programming language to learn?
3. Why use plain text file formats?

Naming files sensible things is good for you and for your computers

WordPress URLs

ROOT/YYYY/MM/DD/words-of-title-separated-by-hyphens

http://cradledincaricature.com/2015/07/24/code-control-and-making-the-argument-in-the-humanities/

The Guardian newspaper:

ROOT/SUB_ROOT/YYYY/MMM/DD/words-describing-content-separated-by-hyphens

http://www.theguardian.com/uk-news/2014/feb/20/rebekah-brooks-rupert-murdoch-phone-hacking-trial

The National Library of Australia's TROVE uses this format:

ROOT/record-type/REF

http://trove.nla.gov.au/work/6315568

The structure of a good file directory:

1. A base or root directory, perhaps called “Committees”.

2. A series of sub-directories by committee name such as “Instruction”, “Discovery”, “Adult Programing” et cetera

3. Within these directories are series of directories for committee reports, meeting minutes, or events.

Introducing a naming convention here that includes a date element keeps the information organised without the need for subdirectories by, say, year or month.

Key Takeaways

1. Data structures should be consistent and predictable.

2. Consider using semantic elements or data identifiers to data directories.

3. Apply naming conventions to directories and file names to identify them, to create associations between data elements, and to assist with the long term readability and comprehension of your data structures.

4. Fit and adapt your data structure to your work.

Pedagogy Check-In

UNIVERSITY OF ALBERTA

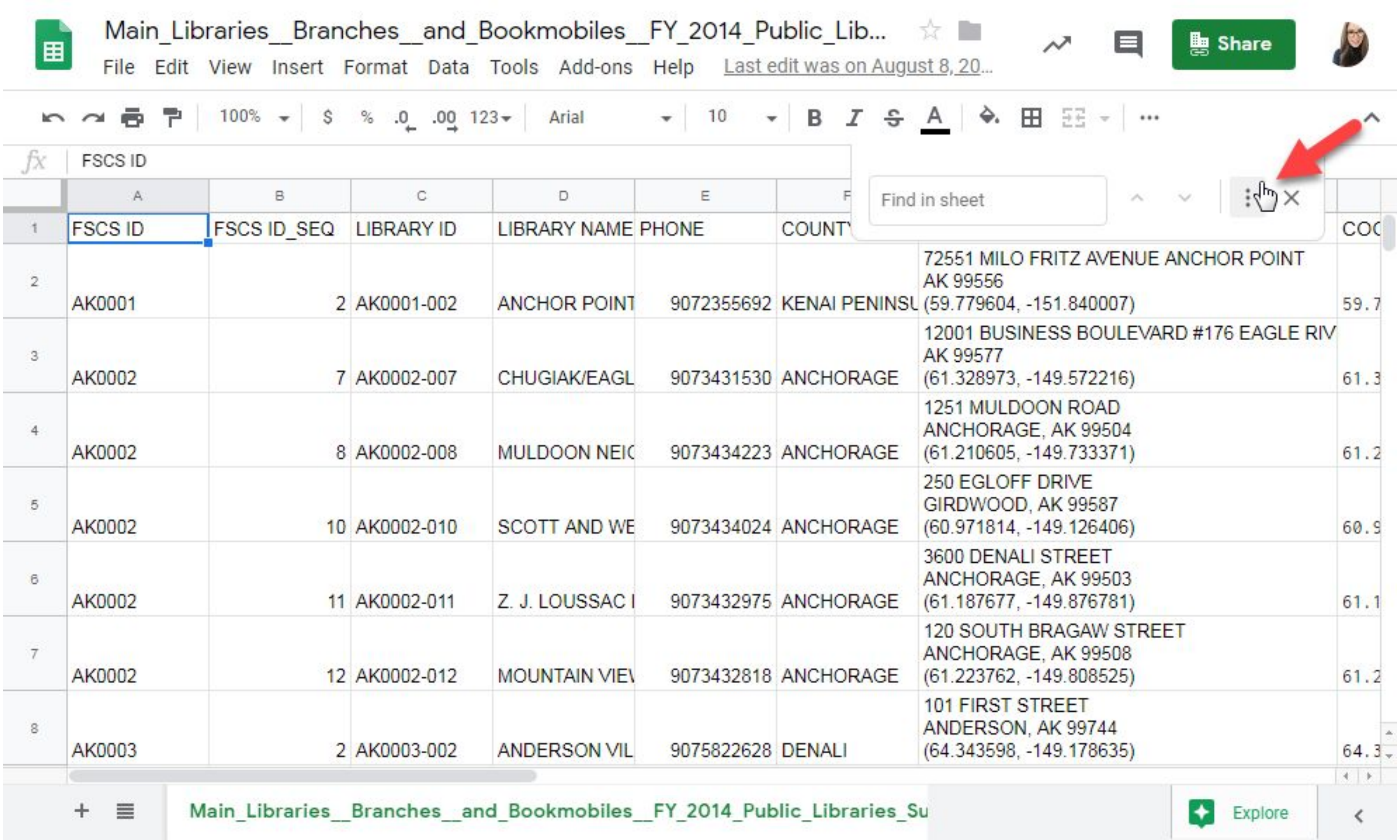
BREAK
15 minutes

Regular Expressions

UNIVERSITY OF ALBERTA

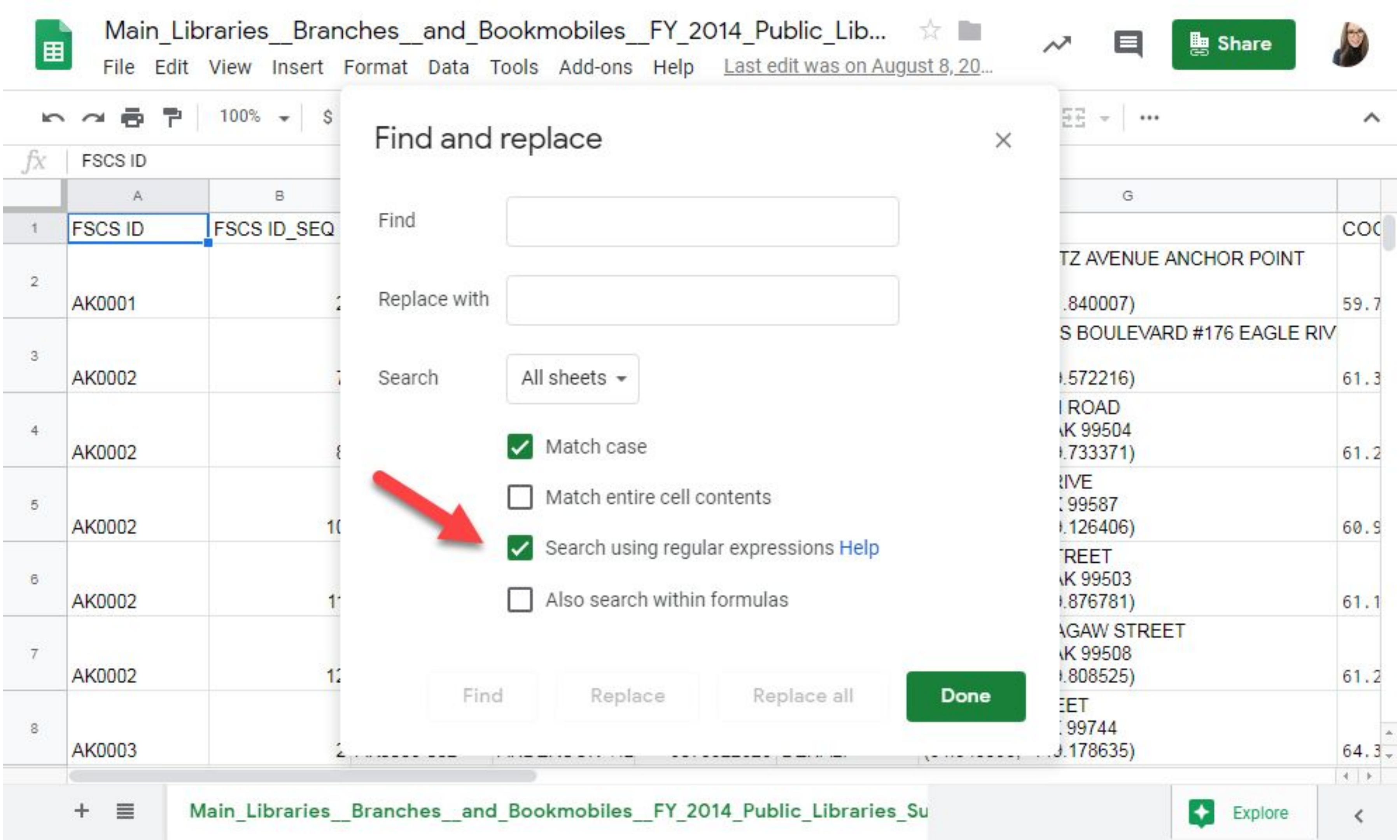
What are Regular Expressions?

- Regular expressions is a language for pattern matching
- Also known as regex
- Will let you:
  - Match on types of characters (e.g. ‘upper case letters’, ‘digits’, ‘spaces’, etc.)
  - Match patterns that repeat any number of times
  - Capture/manipulate the parts of the original string that match your pattern
- Can be used in most programming languages, and tools for working with data like OpenRefine, AntConc, Google Sheets, Excel, some search engines and many more!


Main_Libraries__Branches__and_Bookmobiles__FY_2014_Public_Lib...
File
Edit
View
Insert
Format
Data
Tools
Add-ons
Help
Last edit was on August 8, 20...
Share
100%
$
%
.0
.00
123
Arial
10
B
I
S
A
fx
Find in sheet
Main_Libraries__Branches__and_Bookmobiles__FY_2014_Public_Libraries_Su
Explore

FSCS ID

	A	B	C	D	E	F	G	H
1	FSCS ID	FSCS ID_SEQ	LIBRARY ID	LIBRARY NAME	PHONE	COUNTY		COC
2	AK0001	2	AK0001-002	ANCHOR POINT	9072355692	KENAI PENINSU	72551 MILO FRITZ AVENUE ANCHOR POINT AK 99556 (59.779604, -151.840007)	59.7
3	AK0002	7	AK0002-007	CHUGIAK/EAGL	9073431530	ANCHORAGE	12001 BUSINESS BOULEVARD #176 EAGLE RIV AK 99577 (61.328973, -149.572216)	61.3
4	AK0002	8	AK0002-008	MULDOON NEIC	9073434223	ANCHORAGE	1251 MULDOON ROAD ANCHORAGE, AK 99504 (61.210605, -149.733371)	61.2
5	AK0002	10	AK0002-010	SCOTT AND WE	9073434024	ANCHORAGE	250 EGLOFF DRIVE GIRDWOOD, AK 99587 (60.971814, -149.126406)	60.9
6	AK0002	11	AK0002-011	Z. J. LOUSSAC I	9073432975	ANCHORAGE	3600 DENALI STREET ANCHORAGE, AK 99503 (61.187677, -149.876781)	61.1
7	AK0002	12	AK0002-012	MOUNTAIN VIEV	9073432818	ANCHORAGE	120 SOUTH BRAGAW STREET ANCHORAGE, AK 99508 (61.223762, -149.808525)	61.2
8	AK0003	2	AK0003-002	ANDERSON VIL	9075822628	DENALI	101 FIRST STREET ANDERSON, AK 99744 (64.343598, -149.178635)	64.3



Find and replace

Find

Replace with

Search: All sheets

☑ Match case

☐ Match entire cell contents

☑ Search using regular expressions Help

☐ Also search within formulas

What are Regular Expressions?

- Regular expressions is a language for pattern matching
- Also known as regex
- Will let you:
  - Match on types of characters (e.g. ‘upper case letters’, ‘digits’, ‘spaces’, etc.)
  - Match patterns that repeat any number of times
  - Capture/manipulate the parts of the original string that match your pattern
- Can be used in most programming languages, and tools for working with data like OpenRefine, AntConc, Google Sheets, Excel, some search engines and many more!

A simple regular expression example

Use a regex to locate alternative spellings:

Organi[zs]e

Will find “Organize” or “Organise”

How do regular expressions work?

- Relies on the use of literal characters and metacharacters

Ex. * (asterix)

Literal Use: Looking for the title “The Subtle Art of not Giving a F*ck”

Metacharacter Use: Using research* in a boolean search to locate titles that contain either “research”, “researchers”, “researcher”

Regular Expression Engine

www.regex101.com

Common regex metacharacters

Square brackets can be used to define a list or range of characters to be found.

- [ABC] matches A or B or C
- [A-Z] matches any uppercase letter
- [A-Za-z] matches any upper or lower case letter
    - (note: this is case-sensitive)
- [A-Za-z0-9] matches any upper or lower case letter or any digit
    - (note: this is case-sensitive)

Common regex metacharacters

- . matches any character
- \d matches any single digit
- \w matches any part of word character (equivalent to [A-Za-z0-9])
- \s matches any space, tab, or newline
- \ used to escape the following character when that character is a special character.
- ^ asserts the position at the start of the line.
- $ asserts the position at the end of the line. So what you put before it will only match if they are the last characters of a line.
- \b adds a word boundary. Putting this either side of a word stops the regular expression matching longer variants of words. So:
  - the regular expression library will match and find 666library, library777, 8thlibrary8th
  - the regular expression \blibrary will match library and find library777
  - the regular expression library\b will match library and find 666library
  - the regular expression \blibrary\b will find library but not 666library or library777

Can you guess what the regular expression ^[Oo]rgani.e\b will match?

organise
organize
Organise
Organize
organife
Organike

Or, any other string that starts a line, begins with a letter o in lower or capital case, proceeds with rgani, has any character in the 7th position, and ends with the letter e.

Other useful special characters

- * matches the preceding element zero or more times.
  - For example, ab*c matches “ac”, “abc”, “abbbc”, etc.

- + matches the preceding element one or more times.
  - For example, ab+c matches “abc”, “abbbc” but not “ac”.

- ? matches when the preceding character appears zero or one time.

- {VALUE} matches the preceding character the number of times defined by VALUE; ranges,
  - say, 1-6, can be specified with the syntax {VALUE,VALUE}, e.g. \d{1,9} will match any number between one and nine digits in length.

- | means OR

- Case insensitive setting

Can you guess what the regular expression [Oo]rgani.e\w+$ will match?

organiser
Organized
organifer
Organi2ed111
organisee

Or, any other string that ends a line, begins with a letter o in lower or capital case, proceeds with rgani, has any character in the 7th position, follows with letter e and at least one or more characters from the range [A-Za-z0-9].

[Oo]rgani.e\w+$

Can you guess what the regular expression **^[Oo]rgani.e\w?\b** will match?

organise
Organized
organifer
Organi2ek

Or, any other string that starts a line, begins with a letter o in lower or capital case, proceeds with rgani, has any character in the 7th position, follows with letter e, and ends with zero or one characters from the range [A-Za-z0-9].

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

Can you guess what the regular expression

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

will match?

organise
Organi1e
Organizer
Organifed

Or, any other string that begins with a letter o in lower or capital case after a word boundary, proceeds with rgani, has any character in the 7th position, and end with letter e, or any other string that begins with a letter o in lower or capital case after a word boundary, proceeds with rgani, has any character in the 7th position, follows with letter e, and ends with a single character from the range [A-Za-z0-9].

Exercises

https://librarycarpentry.github.io/lc-data-intro/04-regular-expressions/index.html

Key Takeaways

1. Regular expressions is a language for pattern matching

2. Check your regex with:

   regex101 https://regex101.com/
   rexegper http://regexper.com/
   myregexp http://myregexp.com/
   RegExr http://regexr.com

3. Test yourself with:

   Regex Crossword https://regexcrossword.com/
   Our Multiple Choice Quiz
   https://librarycarpentry.github.io/lc-data-intro/05-quiz/

UNIVERSITY OF ALBERTA

Conclusion

Credits

UNIVERSITY OF ALBERTA

More Questions?

Abigail Sparling *MI*
Serials Metadata Librarian
University of Alberta

@abbisparling

abigail.sparling@ualberta.ca

Céline Gareau-Brennan *MLIS, MBA*
Public Services Librarian,
University of Alberta

@Celine_GB

crg@ualberta.ca