



# Library Carpentry: Tools for Librarians and Humanists

Welcome! Did you install OpenRefine yet?

See the Setup section at the bottom:

<https://gwu-libraries.github.io/2022-07-14-gwu/>



# Welcome!



Dolsy Smith  
Software Development  
Librarian



Leah Richardson  
Special Collections  
Librarian



Josh McDonald  
Collections Strategist

# Helpers

Ricky Graham

Dan Kerchner

Adhithya Kiran

Vakil Smallen



## Thursday

9:00	Introductions and Set-up
9:30	Jargon Busting
10:00	<b>OpenRefine</b>
11:00	Break
11:15	<b>Regular Expressions</b>
12:00	Lunch
1:00	<b>Regular Expressions</b>
2:00	Break
2:15	<b>Python</b>
4:30	END

## Friday

9:00	<b>OpenRefine</b>
10:30	Break
10:45	<b>Python</b>
12:00	Lunch
1:00	<b>Python</b>
2:30	Break
2:45	<b>Python</b>
4:15	Wrap-up
4:30	Post-workshop survey
	END

# Code of Conduct

A welcoming environment for all people is created if we:

- Use welcoming and inclusive language
- Be respectful of different viewpoints and experiences
- Gracefully accept constructive criticism
- Focus on what is best for the community
- Show courtesy and respect towards other community members

More info and how to report concerns:

[https://docs.carpentries.org/topic\\_folders/policies/code-of-conduct.html](https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html)



# Introduce yourself

1. Name and department/major/job
2. What has motivated you to participate in this workshop?
3. What is something that has been an obstacle to your learning in the past?





# Logistics



Pink sticky = I'm stuck!

Green sticky = I'm good!

Yellow sticky = Slow down!

Etherpad for note-taking: [go.gwu.edu/carpentries2022](https://go.gwu.edu/carpentries2022)

Data files to download:

<https://gwu-libraries.github.io/2022-07-14-gwu/>

# Jargon Busting

1. Pair with a neighbor and decide who will take notes on stickies.

2. Write any **terms, phrases, or ideas around code** or software in libraries / digital humanities / text analysis that you have wondered about or you want understand better. One term per sticky.



# Jargon Busting

3. Join with another group. Review the terms each pair came up with. Retain duplicates.

4. Identify common words as a starting point. Spend 10 minutes working together to try to explain what the terms on your list mean.


**Note:** use each other and the Internet as a resource. See Hackpad for some places to look.



1

# Download and Install OpenRefine

<https://openrefine.org/download.html>



OpenRefine 3.4.1 or 3.5.0

Windows users should download

**“Windows kit with embedded Java”**



# What is this dataset?

We will work with metadata for the collection of books that comprise what we know to remain of the early library of the Columbian College (now GW).

459 titles that have “Columbian College in the District of Columbia. Library. former owner.” in their catalog records.

See more here: [go.gwu.edu/cclibrary](https://go.gwu.edu/cclibrary)

**OpenRefine lessons will use the file:**  
columbian-college-ids\_12162021.csv





# OpenRefine Part 1 Agenda

- What is OpenRefine?
- Launching OpenRefine, Importing data, Creating a New Project
- OpenRefine's Interface; Rows vs Records
- View and Edit Data: Facets and Filters
- Edit Data: Clustering
- Columns and Sorting



# What is OpenRefine?

OpenRefine is a desktop application that uses your web browser as a graphical interface.

It is described as “a power tool for working with messy data”

- No internet connection is needed, and none of the data or commands you enter in OpenRefine are sent to a remote server.
- You are NOT modifying original/raw data.
- Projects are autosaved every five minutes and when OpenRefine is properly shut down (Ctrl+C).



# Resources for Working with OpenRefine

[OpenRefine Documentation](#)

[Library Carpentry OpenRefine Curriculum](#)

[Lessons for this workshop](#)

[Lots of tutorials on YouTube](#)





# OpenRefine can be used to standardize and clean data across your file.



Where you have a list of names or terms that differ from each other but refer to the same people, places or concepts.

The data you have:	The data you want:
London : London; London, Londini Londini, [London] : London Londres	London

Subject headings grouped in a single field	SH 1	SH 2	SH 3	SH 4
1700 - 1799; Apologetics--Early works to 1800.; Apologetics--History--18th century.; Free thought--Controversial literature.	1700 - 1799;	Apologetics--Early works to 1800.;	Apologetics--History--18th century.;	Free thought--Controversial literature.

When you have several bits of data combined together in a single column and you want to separate them out with each distinct bit of data into its own column.



# Launch OpenRefine

Common Transformation	Action	GREL expression
To Uppercase	Converts the current value to uppercase	<code>value.toUppercase()</code>
To Lowercase	Converts the current value to lowercase	<code>value.toLowerCase()</code>
To Titlecase	Converts the current value to titlecase (i.e. each word starts with an uppercase character and all other characters are converted to lowercase)	<code>value.toTitlecase()</code>
Trim leading and trailing whitespace	Removes any 'whitespace' characters (e.g. spaces, tabs) from the start or end of the current value	<code>value.trim()</code>



# 2

## Regular Expressions

### Objectives

- ❖ Practice thinking about text computationally
- ❖ Recognize common regex patterns
- ❖ Apply regexes to bibliographic metadata

# Regular Expressions: Imprint statements

```
Oxford Univ
Oxford Univ Pr
Oxford Univ Press
Oxford Univerity Press
Oxford University P
Oxford University Press (UK)
Oxford University Press / UK
Oxford University Press (USA); Clarendon Press
Oxford University Press Inc
Oxford University Press UK
Oxford University Press US
Oxford University Press USA
```

# Regular Expressions: Imprint statements

Londres : B. Bensley, 1821





# Regular Expressions: Imprint statements

Londres : B. Bensley, 1821

(.+): (.), (.+)

# Regular Expressions: Imprint statements

Londres : B. Bensley, 1821

Andover : Flagg and Gould, printers, Published  
and sold by Mark Newman ; 1818

(.+): (.), (.+)

# Regular Expressions: Imprint statements

Londres : B. Bensley, 1821

Andover : Flagg and Gould, printers, Published  
and sold by Mark Newman ; 1818

(.+): (+),  
(\d{4})



# Regular Expressions: Imprint statements

Londres : B. Bensley, 1821

Andover : Flagg and Gould, printers, Published  
and sold by Mark Newman ; 1818

`(.+)` : `(.+)` ; `(\d{4})`

# Regular Expressions: Imprint statements

Londres : B. Bensley, 1821

Andover : Flagg and Gould, printers, Published  
and sold by Mark Newman ; 1818

`(.+)` : `(.+)[,;]` `(\d{4})`

# Regular Expressions: Imprint statements

Londres : B. Bensley, 1821

Andover : Flagg and Gould, printers, Published and sold  
by Mark Newman ; 1818

London, D. Brown, 1722

`(.+)` : `(.+)[,;]` `(\d{4})`



# Regular Expressions: Imprint statements

Londres : B. Bensley, 1821

Andover : Flagg and Gould, printers, Published and sold  
by Mark Newman ; 1818

London, D. Brown, 1722

`(.+)` `[: ,]` `(.+)` `[ , ;]` `(\d{4})`

# Regular Expressions: Imprint statements

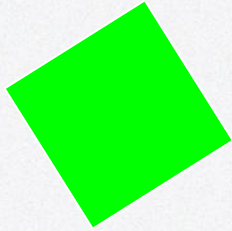
Londres : B. Bensley, 1821

Andover : Flagg and Gould, printers, Published and sold  
by Mark Newman ; 1818

London, D. Brown, 1722

`(.+)\s?[: ,] (.+) [, ;] (\d{4})`

# Lunchtime feedback



1. Green sticky: What is something that is working well for you in this workshop?



2. Pink sticky: What is a question you still have about OpenRefine or regular expressions?





**3**

# Regular Expressions

# Regular Expressions: Exercise

1. Philadelphia, Printed by William W. Woodward,  
1817-1819
2. Boston : Printed by Peter Edes, in State-Street,  
[1785]
3. Newburyport [Mass.] : Printed by Angier March, 1802
4. Washington, D.C. : Judd & Detweiler, printers, 1870
5. Boston : Printed by Samuel Etheridge for J. White,  
Thomas and Andrews, E. Larkin, W.P. Blake, J. West  
and J. Boyle, MDCCXCV [1795]



4

# Python

<https://colab.research.google.com>

Start a **new notebook**





# Objectives

- Explore the Python interpreter via a Jupyter notebook
- Create variables to hold different types of data
- Read data in from a file
- Transform data from one type to another
- Work with conditionals and iterative structures to process data efficiently
- Store output in a persistent and portable format



# Python exercise #1.1

Execute the following commands, each in its own code cell in your notebook.

```
In [ ]: "Hello world"
```

```
In [ ]: 9 + 3
```

```
In [ ]: "9 + 3"
```

```
In [ ]: Hello world
```

```
In [ ]: Hello
```

```
In [ ]: file_name = 'newton_opticks.txt'
```

```
In [ ]: print(file_name)
```



# Python exercise #1.1

What kinds of input are valid for Python?

What do you notice about the output you receive?





# Python exercise #1.4

Run the following lines of code and discuss with your neighbor. What do these operations do?

```
In [ ]: text[:100]
```

```
In [ ]: len(text)
```

```
In [ ]: text.split()
```

# Python exercise #1.5

Repeat the above operations with our `words` variable. How do lists behave differently from strings?

```
In [ ]: words[0]
```

```
In [ ]: words[:100]
```

```
In [ ]: len(words)
```

```
In [ ]: words.split()
```

# Python exercise #1.6

How does `word == 'light'` in our little program define truth?

Are there situations where our test would *not* catch instances that we might want to count as true?

Are there situations where it might flag as true instances that we would want to count as false?





# Python exercise #1.7

Can you modify the previous `for` loop to use the regular expression to check for the presence of the word `light` in Newton's text?

Use the link in the Etherpad to access the exercise.



# Python exercise #1.8

How would we create a table showing the frequency of occurrence of every word in the document?

Working with a neighbor, develop a logical plan for this task, using the `words` list we created above.



# Python exercise #1.9

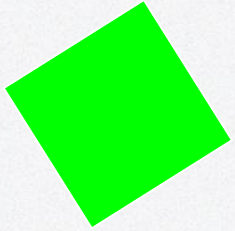
Let's try to put together loops, conditionals, dictionaries, and lists in order to create a dictionary of word frequencies in Newton's *Opticks*.

See if you can put the lines of code in the correct order, using the link to Exercise 9 on the Etherpad.





# End of Thursday feedback



Green sticky: What is something you learned today?



Pink sticky: What is something we could improve for tomorrow?



# Library Carpentry Day 2

Get ready for OpenRefine lesson

## Thursday

9:00	Introductions and Set-up
9:30	Jargon Busting
10:00	<b>OpenRefine</b> Part 1
11:00	Break
11:15	<b>Regular Expressions</b> Part 1
12:00	Lunch
1:00	<b>Regular Expressions</b> Part 2
2:00	Break
2:15	<b>Python</b> Part 1
4:30	END

## Friday

9:00	<b>OpenRefine</b> Part 2
10:30	Break
10:45	<b>Python</b> Part 2
12:00	Lunch
1:00	<b>Python</b> Part 2-3
2:30	Break
2:45	<b>Python</b> Part 3
4:15	Wrap-up
4:30	Post-workshop survey
	END



The slide features four decorative geometric patterns in the corners, each composed of overlapping squares and triangles in various colors. Top-left: Green and dark grey. Top-right: Yellow and dark grey. Bottom-left: Red and dark grey. Bottom-right: Teal and dark grey.

**5**

# OpenRefine

# OpenRefine Part 2 Agenda

- Introduction to Common Transformations
- Undo and Redo
- Writing Transformations
- Transforming Strings, Numbers, Dates and Booleans with GREL and RegEx
- Handling Arrays
- Going Further using Regular Expressions in OpenRefine (time permitting)
- Exporting transformed data





**6**

**Python**



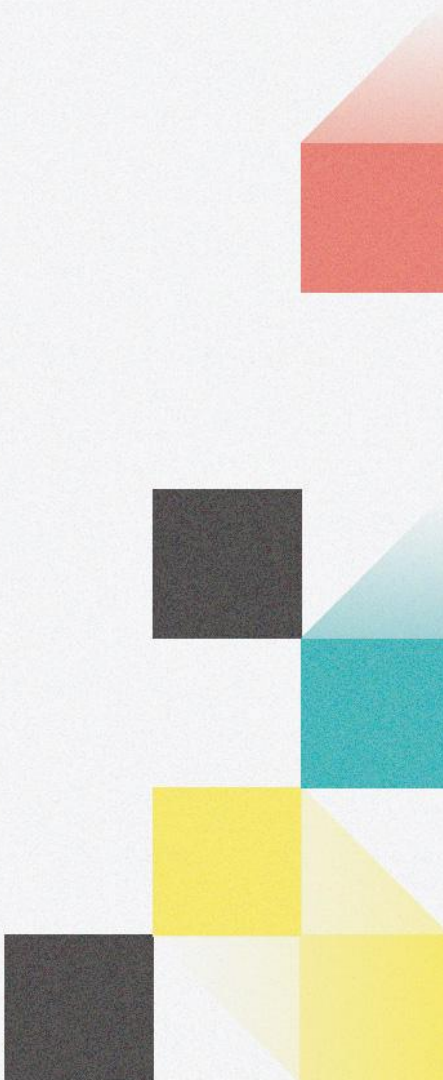


# Python exercise 2.1

1. Create a subset of volumes where the language is Latin.

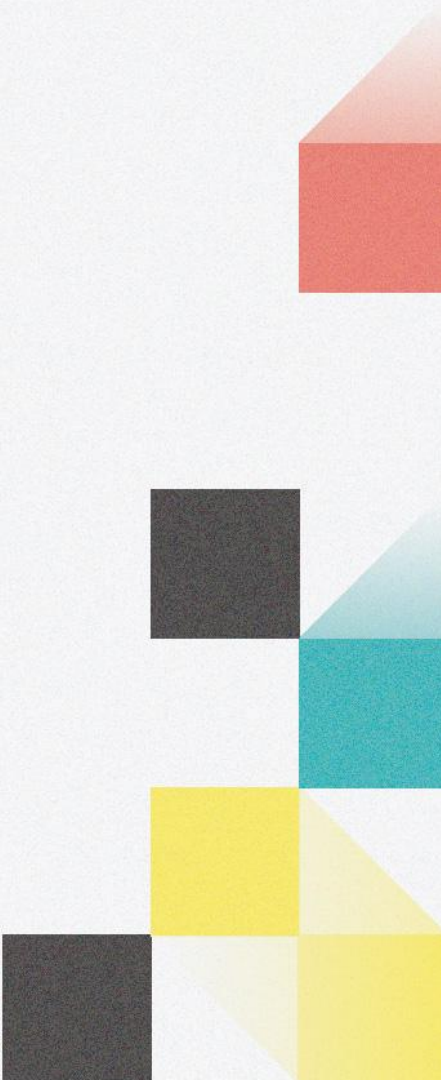
2. Create a subset of volumes where the language is Latin AND the author is Isaac Newton.

Hints:

- Put each condition in parentheses.
  - In pandas, you use the | (pipe) and & (ampersand) characters to represent OR and AND.
- 



# Python exercise 2.2

1. Determine who the most frequent authors (or "Creator"s) are in the set.
  2. Make a plot showing the top 25 authors. Experiment with the formatting.
- 



**7**

**Python**



# Objectives

- Explore HathiTrust extracted-feature datasets
- Work with core elements for the computational analysis of texts, including
  - Tokens
  - Parts of speech
  - Collocations
- Practice using the pandas library to query, sort, and group datasets



# Python exercise #3.1

Visit the HathiTrust Catalog and look for the edition of Newton's *Opticks* published by William Innys in 1730.

**<https://www.hathitrust.org/>**

# Python exercise #3.2

Compare the token counts with those you obtained in our Python lesson on Day 1.

How does your tokenization differ from what's represented here?



# Python exercise #3.3

Use the `term_counts` DataFrame to find out how many times the token `light` appears in this text.

You can find the meaning of the `pos` (part of speech) tags on the Penn Treebank website (see the link on the Etherpad).

# Python exercise #3.5

Can you recreate the kind of token count we got from `term_volume_freqs` by using the `groupby` method on the page-level DataFrame (`df`)?

# Python exercise #3.6

Find a volume in HathiTrust that interests you. See if you can replicate the steps above to do the following:

1. Load the extracted features dataset.
2. Find the most common tokens.
3. Find the most common noun tokens (or some other part of speech).
4. Pick a particular token and find which other tokens occur most commonly with it on the same page.



# Wrap-Up

- Links to lesson materials
- Post-workshop survey
- Coding consultations:  
<https://calendly.com/gwul-calendly>
- More workshops

