

Working with Bibliographic Data I

Dr Christopher Ohge
15.03.2023



**LONDON RARE
BOOKS SCHOOL**
INSTITUTE OF ENGLISH STUDIES
UNIVERSITY OF LONDON

Plan for today

1. What is ‘software carpentry’, and why should you care about *library carpentry*?
2. Principles of Working with (Unstructured) Text Data
 - a. Command Line Interface
 - b. Regular Expressions
 - c. Grep
3. Git for version control
4. Working with data:
 - a. Data Visualisation Exercise: Rawgraphs
 - b. OpenRefine

Download [sample dataset](#) from the [Universal Short Title Catalog](#) (USTC) project.

Software Carpentry

Software carpentry teaches data fundamentals using an ***applied approach***, ***avoiding the theoretical and general in favor of the practical and specific***, by *showing learners how to solve specific problems with specific tools and providing hands-on practice.*

Library carpentry focuses on building software and data skills within library and information-related communities. *The goal is to empower people in these roles to use software and data in their own work and to become advocates for and train others in efficient, effective, and reproducible data and software practices.*

Data and data structure as source study

Tim Hitchcock: 'We can now engage with that new resource in new ways, but to do so we need scholars who can work with old sources in their new digital guide, and who can recognise that digital makes them different'.

Digital format of records can be understood as a new kind of historical source study.

Born-digital versus digitised data:

- Born-digital created from (and for) digital applications (rarely intended for humanistic study, therefore messy)
- Digitised data is converted from printed sources (good for humanistic study but often unclear what methods were used in the conversion)

Plain text

‘Plain text’ refers to platform-independent text data that can be parsed and processed by the computer.

File extension	Format	Plain text?
.doc, .docx	Word processing	no
.xls, .xlsx	spreadsheets	no
.txt	Any text	yes
.xml	extensible (structured) markup	yes
.csv, .tsv	Comma-separated or tab-separated values for spreadsheets	yes

(Jonathan Blaney, Sarah Milligan, Marty Steer, and Jane Winters, *Doing Digital History*, Manchester University Press, 2021.)

Command Line Interface (CLI)

CLI primarily allows you to navigate through your computer and to run scripts for a variety of purposes.

Most people interact with personal computers is called a **graphical user interface** (GUI), but CLI allows you to use commands to do specific tasks (rather than through menu-based items).

The program for performing CLI work is usually a **unix shell** or **terminal**. A unix shell is both a **command-line interface** (CLI) to read commands, and a scripting language that allows repetitive tasks to be done automatically and fast.

The most popular Unix shell is Bash (which is standard on most Mac- and Linux-based systems).

Bash shell prompt

When you first open the shell, you will see a **prompt**, indicating that the shell is waiting for input. Usually you will see the prompt begin with a dollar sign (\$).

You then add the first command.

WINDOWS COMMAND

run ---> cmd

- cd cd Desktop\Test
- dir
- del
- copy
- move
- rename
- mkdir
- rmdir
- cd
- help

OSX TERMINAL / UNIX

Launchpad ---> Terminal

- cd cd Desktop/Test
- ls
- rm
- cp
- mv
- rename: mv
- mkdir
- rmdir
- pwd
- man

Version control with Git

Git is a version control software.

Most people use the cloud-based tool [GitHub](#) to host data, collaborate with others, and manage and share their work.

Git allows users to access repository data from the cloud (via a **pull**), make changes, add and **commit** those changes, and **push** the changed data back to the cloud so that other collaborators can see the changes.

Learn more about Git

[Software Carpentry Tutorial](#)



Regular expressions

- Sequence of characters that define a search pattern
- Powerful way of searching and manipulating data

`(?<=\\.) {2,}(?=[A-Z])`

I watch three climb before it's my turn. It's a tough one. The guy before me tries twice. He falls twice. After the last one, he comes down. He's finished for the day. It's my turn. My buddy says "good luck!" to me. I noticed a bit of a problem. There's an outcrop on this one. It's about halfway up the wall. It's not a

Regular expressions

[ChBl]eating matches *Cheating* and *Bleating*

Regular expressions

`[0123456789]` matches any number

Regular expressions

[0-9] matches any number

Regular expressions

`[0-9|]` matches any number or `/`

So `[0-9|]+` matches *1717* or */7/7*

Regular expressions

What does `[a-z]+` match?

Demo

<https://regex101.com>

<http://txti.es/bleak-housexml>

The screenshot shows the regex101.com website interface. At the top, there's a navigation bar with the site name, social media links, and utility links like 'donate', 'contact', 'bug reports & feedback', and 'wiki'. The main area is divided into three sections: 'REGULAR EXPRESSION', 'TEST STRING', and 'EXPLANATION'. The 'REGULAR EXPRESSION' section shows the pattern `. *? \.` with a status of '15 matches, 2051 steps (-53ms)'. The 'TEST STRING' section contains an XML document snippet: `<?xml version="1.0" encoding="UTF-8"?> <text>Bleak House Chapter 1` followed by a paragraph of text from 'Bleak House'. The 'EXPLANATION' section provides a detailed breakdown of the regex components: `.` matches any character except line terminators, `*?` is a quantifier for zero or one match (lazy), `\.` matches the literal character dot, and it lists global pattern flags `g` (global) and `m` (multi-line). Below the explanation, the 'MATCH INFORMATION' section lists three matches: Match 1 (0-17) for the XML declaration, Match 2 (81-88) for the word 'LONDON.', and Match 3 (88-172) for the start of the first paragraph.

regular expressions 101

@regex101 donate contact bug reports & feedback wiki

REGULAR EXPRESSION 15 matches, 2051 steps (-53ms)

TEST STRING SWITCH TO UNIT TESTS

`<?xml version="1.0" encoding="UTF-8"?> <text>Bleak House Chapter 1`

In Chancery

LONDON. Michaelmas Term lately over, and the Lord Chancellor sitting in Lincoln's Inn Hall. Implacable November weather. As much mud in the streets as if the waters had but newly retired from the face of the earth, and it would not be wonderful to meet a Megalosaurus, forty feet long or so, waddling like an elephantine lizard up Holborn Hill. Smoke lowering down from chimney-pots, making a soft black drizzle, with flakes of soot in it as big as full-grown snow-flakes – gone into mourning, one might imagine, for the death of the sun. Dogs, undistinguishable in mire. Horses, scarcely better; splashed to their very blinkers. Foot passengers, jostling one another's umbrellas in a general infection of ill-temper, and losing their foot-hold at street-corners, where tens of thousands of other foot passengers have been slipping and sliding since the day broke (if the day ever broke). adding new deposits to

EXPLANATION

- `.` matches any character (except for line terminators)
- `*?` Quantifier — Matches between **zero** and **unlimited** times, as few times as possible, expanding as needed (**lazy**)
- `\.` matches the character `.` literally (case sensitive)
- Global pattern flags
 - `g` modifier: global. All matches (don't return after first match)
 - `m` modifier: multi line. Causes `^` and `$` to match

MATCH INFORMATION

Match 1

Full match 0-17 `<?xml version="1.`

Match 2

Full match 81-88 `LONDON.`

Match 3

Full match 88-172 `Michaelmas Term lately ov`

Examples

<https://regex101.com>

<http://txti.es/bleak-housexml>

Regular expressions to:

- Match all sentences
- Match all words
- Match all words at end of sentence
- Match all words start with 's'
- Match all words ending with 's'
- Match all Proper Nouns

The screenshot shows the regex101.com website interface. The top navigation bar includes the site name, a user profile icon, and links for donate, contact, bug reports, and feedback. The main content area is divided into three sections: 'REGULAR EXPRESSION', 'TEST STRING', and 'EXPLANATION'. The 'REGULAR EXPRESSION' section shows the pattern `.*?\.` with a status of '15 matches, 2051 steps (-53ms)'. The 'TEST STRING' section contains XML content from 'Bleak House' by Charles Dickens, with several lines highlighted in blue to indicate matches. The 'EXPLANATION' section provides a detailed breakdown of the pattern: `.*?` matches any character (except for line terminators) zero or more times, expanding as needed (lazy); `\.` matches the character `.` literally (case sensitive). It also lists global pattern flags: `g` (global) and `m` (multi line). The 'MATCH INFORMATION' section lists three matches: Match 1 (Full match 0-17: `<?xml version="1.`), Match 2 (Full match 81-88: `LONDON.`), and Match 3 (Full match 88-172: `Michaelmas Term lately ov`).

Solutions

<https://regex101.com>

<http://txti.es/bleak-housexml>

Regular expressions to:

- Match all sentences
- Match all words
- Match all words at end of sentence
- Match all words start with 's'
- Match all words ending with 's'
- Match all Proper Nouns

`. *? \`
`.`

`\b . *? \`
`b`

`\w *? \`
`.`

`\b s . *? \`
`b`

`[A-Za-z]+s`

`[A-Z][a-z]*? \b`

regular expressions 101

@regex101 donate contact bug reports & feedback wiki

REGULAR EXPRESSION 15 matches, 2051 steps (~53ms)

TEST STRING SWITCH TO UNIT TESTS

`<?xml version="1.0" encoding="UTF-8"?> <text>Bleak House Chapter 1`

In Chancery

LONDON. Michaelmas Term lately over, and the Lord Chancellor sitting in Lincoln's Inn Hall. Implacable November weather. As much mud in the streets as if the waters had but newly retired from the face of the earth, and it would not be wonderful to meet a Megalosaurus, forty feet long or so, waddling like an elephantine lizard up Holborn Hill. Smoke lowering down from chimney-pots, making a soft black drizzle, with flakes of soot in it as big as full-grown snow-flakes – gone into mourning, one might imagine, for the death of the sun. Dogs, undistinguishable in mire. Horses, scarcely better; splashed to their very blinkers. Foot passengers, jostling one another's umbrellas in a general infection of ill-temper, and losing their foot-hold at street-corners, where tens of thousands of other foot passengers have been slipping and sliding since the day broke (if the day ever broke). adding new deposits to

EXPLANATION

`. *? \` / gm

- `. *? \` matches any character (except for line terminators)
- `*? Quantifier` — Matches between **zero** and **unlimited** times, as few times as possible, expanding as needed (**lazy**)
- `\` matches the character `\` literally (case sensitive)
- Global pattern flags**
 - g modifier:** global. All matches (don't return after first match)
 - m modifier:** multi line. Causes `^` and `$` to match

MATCH INFORMATION

Match 1

Full match 0-17 `<?xml version="1.`

Match 2

Full match 81-88 `LONDON.`

Match 3

Full match 88-172 `Michaelmas Term lately ov`

Regex practice

- <https://regexone.com/>

Notes

- `ar` – search for string you want to find
- `a.l` - wildcards or character classes, the dot
- `a.\.` – to find a fullstop (escape with backslash)
- `{5}` - quantifiers – number of repetitions, not overlapping
- `|{2}`
- `.` * or `+` - all or one or more
- `m[aeiou]` – any one character, say vowels, or digits `[0-9]`
- `m[0-9]+`
- `[A-Z][aeiou]`
- `\w` – any alphanumeric (or `\W` non alpha)
- `\d` – any digit (`\D` non digit)
- `\s` – any whitespace (space, tab, newline) (or `\S` non whitespace)
- `[a-z]+?s\W`
- `t.*s` - all words begin with T and end with S, then introduce greedy ?
- `\b[Tt]\S+?s\b`– word break, start with T, end with S, word break

grep

- `grep` stands for global regular expressions. This is a powerful command to search through multiple files. It literally processes a text line by line and prints the results that match the regular expression.
- Syntax:
`$ grep [OPTIONS] PATTERN [FILE...]`
- On the command line, use `cd` to navigate to "corpus" (i.e., the folder you just downloaded).

grep and sed

- The next major use of regex on the command line is `sed` (literally "stream editor"), a simple search and replace function that can be used for multiple files. Recall our problem with the word "mad'st". With `sed` we can regularise the word over multiple files using this syntax:

```
$ sed -e "s/STRING_TO_REPLACE/STRING_TO_REPLACE_IT/g" file.txt
```

RawGraphs

RawGraphs is free, open source, web-based data visualization software that can create and export more than 30 types of data visualisations.

Go to the SAS PORT page to access the [RawGraphs exercise](#).

The logo for RawGraphs, featuring the word "RAW" in bold black uppercase letters and "Graphs" in a green sans-serif font, with a light green circular glow behind the text.

OpenRefine

[OpenRefine](#) is a free, open source tool for working with data: cleaning it; transforming it from one format into another; and extending it with web services and external data.

Typical uses:

1. Remove duplicate records
2. Separate multiple values contained in the same field
3. Analyse the distribution of values throughout a data set
4. Group together different representations of the same reality



OpenRefine

Learn more about OpenRefine

[Programming Historian tutorial](#)

Programming Historian

ABOUT ▾

CONTRIBUTE ▾

LESSONS

EVENTS

SUPPORT US ▾

BLOG

EN

ES

FR

PT



Cleaning Data with OpenRefine

Seth van Hooland, Ruben Verborgh, and Max De Wilde

This tutorial focuses on how scholars can diagnose and act upon the accuracy of data.

 Peer-reviewed

 CC-BY 4.0

 Support PH