Working with Data: Regular Expressions

R, Regular Expressions & SQL DUSC course (December 7th 2023)

Instructor: Katie Falà

https://librarycarpentry.org/lc-data-intro/01-regular-expressions.html

THE CARPENTRIES

Introductions

- Instructor: Katie Falà (she/they)

  Research Fellow in
  Microbial Bioinformatics
  fala.katie@gmail.com
  @forestsomewhere

- Helpers:
  - Aleksandra (Ola)
  - Diego
- Host:
  - Lucie

Schedule

10:00-10:15 Intro

10:15 -11:00 Introduction to Regular expressions

11:00-11:10 Coffee break 1 ☕

11:10-12:00 Regular expressions 2

12:00-13:00 Lunch break 🍽️

13:00-14:00 Matching & Extracting Strings 1

14:00-14:10 Coffee break 2 ☕

14:10-15:00 Matching & Extracting Strings 2

Practicalities

- Etherpad: https://pad.carpentries.org/2023-12-05-dusc-r-regex-sql
- The Carpentries Code of Conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html
  - Show courtesy and respect towards other community members
- No specialised software needed today:
  - Zoom, Browser

Online learning can be difficult!

- Expect interaction, but in whichever form works for you
- Verbal/Non-verbal feedback (typing comments, reactions on Zoom)
- Questions/issues: Raise hand/type in Zoom chat or Etherpad, message one of the helpers or myself directly

Schedule

~~10:00-10:15 Intro~~

10:15 -11:00 Introduction to Regular expressions

11:00-11:10 Coffee break 1 ☕

11:10-12:00 Regular expressions 2

12:00-13:00 Lunch break 🍽️

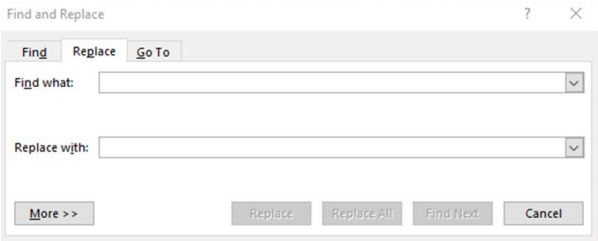
13:00-14:00 Matching & Extracting Strings 1

14:00-14:10 Coffee break 2 ☕

14:10-15:00 Matching & Extracting Strings 2

Introduction to Regular Expressions (RegEx)

- Build a sequence of characters to define a search for matching strings
- Simple string searches:
  - Words e.g. “Edinburgh”
  - Numbers e.g. “2023”
  - Alphanumeric e.g. National Insurance Numbers “AA111111A.”



Wildcard characters in library searches:

E.g. Searching for “Colo?r” will match both color and colour.

Introduction to Regular Expressions

- RegEx allow you to perform **more sophisticated searches**, finding strings or files that match a **pattern**, rather than specific strings:
  - Case: “Edinburgh”, “edinburgh”, “EDINGBURGH”
  - Misspellings: “Edinburgh”, “Eedingburgh”, “edinburg”, “edinbourgh”
  - Matching different date formats e.g. “07-12-2023” and “07/Dec/2023”
  - Matching all phone numbers with the pattern “(+XX) XXXXXXXXX
- Amplify your capacity to find, manage, and transform your data and files.

- RegEx are a general concept, originally developed in the 1950s, for which several implementations (“flavours”) exist.
- Most implementations employ similar syntaxes and metacharacters and behave similarly for most pattern-matching, but have subtle differences

Building regular expressions

Literal Characters + **Meta-characters** → **RegEx!**

ABC
Abc
0123

Any American Standard Code for Information Interchange (ASCII) character that has a special meaning

Find strings or files that match a pattern, rather than a specific string

Literal characters

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.
- **`[A-Za-z0-9]`** matches any upper or lower case letter or any digit.

Common regex metacharacters

- . matches any character.
- **\d** matches any single digit (equivalent to [0-9]).
- **\w** matches any word character, including underscores (equivalent to [A-Za-z0-9_]).
- **\s** matches any space, tab, or newline.

Escaping special characters

- What if we wanted to find matches of website addresses containing ".com"?

➢ Problem: "." is itself a special character (matches any character)
➢ Strategy: "\" used to escape the following character when that character is a special character.
➢ Solution: A regular expression to find .com would be \.com

“Anchors” - positional metacharacters

- ^ (“caret” or “circumflex”) asserts the position at the start of the line.
  - What you put after the caret will only match if they are the first characters of a line.
- $ asserts the position at the end of the line.
  - What you put before it will only match if they are the last characters of a line.
- \b asserts that the pattern must match at a word boundary. Putting this either side of a word stops the regular expression matching longer variants of words.

Word boundaries

- the regular expression mark will match not only mark but also find marking, market, unremarkable, and so on.
- the regular expression \bword will match word, wordless, but not reword.
- the regular expression comb\b will match comb and honeycomb but not combine.
- the regular expression \brespect\b will match respect but not respectable or disrespectful

Exercise 1.1

What will the regular expression

“^[Oo]rgani.e\b”

match?

Cheat Sheet

`[ABC]` : matches A or B or C.

`[A-Z]` : matches any uppercase letter.

`[A-Za-z]` : matches any upper or lower case letter.

`[A-Za-z0-9]` : matches any upper or lower case letter or any digit.

`.` : 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.

`\` : escape the following special character

`^` : only match if they are the first characters of a line

`$` : only match if they are the last characters of a line.

`\b` : only match at a word boundary.

Exercise 1.1

What will the regular expression

"^[Oo]rgani.e\b"

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or any digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

Exercise 1.1

What will the regular expression

"^[Oo]rgani.e\b"

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or any digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

Exercise 1.1

What will the regular expression

"^[Oo]rgani.e\b"

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or any digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

Exercise 1.1

What will the regular expression

"^[Oo]rgani.e\b"

match?

Let's test our understanding:

PollEverywhere Link

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or any digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

Exercise 1.1

What will the regular expression

"^[Oo]rgani.e\b"

match?

✔
Organise
organi3e

✘
They organise
Organizer

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or any digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

Don’t panic!

- We <u>are</u> programming here
- But rather than inventing new symbols/variables as would be done in other languages, we are using combinations of standard ASCII keys (metacharacters) to achieve our goals
- RegEx is not intuitive, but you will become more capable with practice
- Open book: keep Cheat Sheet at hand

Schedule

~~10:00-10:15 Intro~~

~~10:15 -11:00. Introduction to Regular expressions 1~~

11:00-11:10 Coffee break 1 ☕

11:10-12:00 Introduction to Regular expressions 2

12:00-13:00 Lunch break 🍽️

13:00-14:00 Regular expressions 3

14:00-14:10 Coffee break 2 ☕

14:10-15:00 Regular expressions 4

Other useful special characters: “repeaters”

- `*` 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.
  - For example, ab?c matches “ac”, “abc”, but not “abbc”.
- `{VALUE}` matches the preceding character the number of times defined by VALUE;
  - ab{3}c matches “abbbc”, but not “abc”, “abbc”
  - You can specify a range for the number of times with the syntax `{VALUE,VALUE}`,
  - e.g. `\d{1,9}` will match any number between one and nine digits in length.
- | means or.
- /i renders an expression case-insensitive (equivalent to [A-Za-z]).

Exercise 1.2

What will the regular expression

“^[Oo]rgani.e\w*”

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

***** matches the preceding element zero or more times.

+ matches the preceding element one or more times.

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

{VALUE} matches the preceding character the number of times defined by VALUE;

| : matches either/or.

/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.2

What will the regular expression

“^[Oo]rgani.e\w*”

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

***** matches the preceding element zero or more times.

+ matches the preceding element one or more times.

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

{VALUE} matches the preceding character the number of times defined by VALUE;

| : matches either/or.

/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.2

What will the regular expression

“^[Oo]rgani.e\w*”

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

***** matches the preceding element zero or more times.

+ matches the preceding element one or more times.

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

{VALUE} matches the preceding character the number of times defined by VALUE;

| : matches either/or.

/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.2

What will the regular expression

"^[Oo]rgani.e\w*"

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

***** matches the preceding element zero or more times.

+ matches the preceding element one or more times.

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

{VALUE} matches the preceding character the number of times defined by VALUE;

| : matches either/or.

/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.2

What will the regular expression

"^[Oo]rgani.e\w*"

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

***** matches the preceding element zero or more times.

+ matches the preceding element one or more times.

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

{VALUE} matches the preceding character the number of times defined by VALUE;

| : matches either/or.

/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.2

What will the regular expression

"^[Oo]rgani.e\w*"

match?

Test your understanding:

PollEverywhere Link

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

***** matches the preceding element zero or more times.

+ matches the preceding element one or more times.

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

{VALUE} matches the preceding character the number of times defined by VALUE;

| : matches either/or.

/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.2

What will the regular expression

"^[Oo]rgani.e\w*"

match?

✔
organise
organi2ed111
Organi3e
Organised
Organizer
organise**e**tion
Organi2ed111

✘
They organize
Organis**a**tion

Cheat Sheet

[ABC] : matches A or B or C.
[A-Z] : matches any uppercase letter.
[A-Za-z] : matches any upper or lower case letter.
[A-Za-z0-9] : matches any upper or lower case letter or digit.
. : 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.
**** : escape the following special character
^ : only match if they are the first characters of a line
$: only match if they are the last characters of a line.
\b : only match at a word boundary.
***** matches the preceding element zero or more times.
+ matches the preceding element one or more times.
? matches when the preceding character appears zero or one time.
{VALUE} matches the preceding character the number of times defined by VALUE;
| : matches either/or.
/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.3

What will the regular expression

“[Oo]rgani.e\w+$”

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

***** matches the preceding element zero or more times.

+ matches the preceding element one or more times.

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

{VALUE} matches the preceding character the number of times defined by VALUE;

| : matches either/or.

/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.3

What will the regular expression

“[Oo]rgani.e\w+$”

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

***** matches the preceding element zero or more times.

+ matches the preceding element one or more times.

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

{VALUE} matches the preceding character the number of times defined by VALUE;

| : matches either/or.

/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.3

What will the regular expression

“[Oo]rgani.e\w+$”

match?

✔
Organised
Organizer
organisetion
Organi2ed111
She organised

✘
She organised it
organise

Cheat Sheet

[ABC] : matches A or B or C.
[A-Z] : matches any uppercase letter.
[A-Za-z] : matches any upper or lower case letter.
[A-Za-z0-9] : matches any upper or lower case letter or digit.
. : 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.
**** : escape the following special character
^ : only match if they are the first characters of a line
$: only match if they are the last characters of a line.
\b : only match at a word boundary.
***** matches the preceding element zero or more times.
+ matches the preceding element one or more times.
? matches when the preceding character appears zero or one time.
{VALUE} matches the preceding character the number of times defined by VALUE;
| : matches either/or.
/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.4

What will the regular expression

“^[Oo]rgani.e\w?\b”

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

***** matches the preceding element zero or more times.

+ matches the preceding element one or more times.

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

{VALUE} matches the preceding character the number of times defined by VALUE;

| : matches either/or.

/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.4

What will the regular expression

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

match?

✔
Organised
Organizer
Organise
Organised it

✘
She organised
She organised it
Organi2edd
Organi2ed1

Cheat Sheet

[ABC] : matches A or B or C.
[A-Z] : matches any uppercase letter.
[A-Za-z] : matches any upper or lower case letter.
[A-Za-z0-9] : matches any upper or lower case letter or digit.
. : 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.
**** : escape the following special character
^ : only match if they are the first characters of a line
$: only match if they are the last characters of a line.
\b : only match at a word boundary.
***** matches the preceding element zero or more times.
+ matches the preceding element one or more times.
? matches when the preceding character appears zero or one time.
{VALUE} matches the preceding character the number of times defined by VALUE;
| : matches either/or.
/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.5

What will the regular expression

“^[Oo]rgani.e\w?$”

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

***** matches the preceding element zero or more times.

+ matches the preceding element one or more times.

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

{VALUE} matches the preceding character the number of times defined by VALUE;

| : matches either/or.

/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.5

What will the regular expression

"^[Oo]rgani.e\w?$"

match?

✔
Organised
Organizer
Organise

✘
She organised
She organised it
She organised it
Organi2edd
Organi2ed1

Cheat Sheet

[ABC] : matches A or B or C.
[A-Z] : matches any uppercase letter.
[A-Za-z] : matches any upper or lower case letter.
[A-Za-z0-9] : matches any upper or lower case letter or digit.
. : 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.
**** : escape the following special character
^ : only match if they are the first characters of a line
$: only match if they are the last characters of a line.
\b : only match at a word boundary.
***** matches the preceding element zero or more times.
+ matches the preceding element one or more times.
? matches when the preceding character appears zero or one time.
{VALUE} matches the preceding character the number of times defined by VALUE;
| : matches either/or.
/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.6

What will the regular expression

“\b[Oo]rgani.e\w{2}\b”

match?

Cheat Sheet

[ABC] : matches A or B or C.

[A-Z] : matches any uppercase letter.

[A-Za-z] : matches any upper or lower case letter.

[A-Za-z0-9] : matches any upper or lower case letter or digit.

. : 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.

**** : escape the following special character

^ : only match if they are the first characters of a line

$: only match if they are the last characters of a line.

\b : only match at a word boundary.

***** matches the preceding element zero or more times.

+ matches the preceding element one or more times.

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

{VALUE} matches the preceding character the number of times defined by VALUE;

| : matches either/or.

/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.6

What will the regular expression

“\b[Oo]rgani.e\w{2}\b”

match?

✔
Organisers
Organised1
organisedd
She organisedd

✘
Organise
She organised
Organiseddd

Cheat Sheet

[ABC] : matches A or B or C.
[A-Z] : matches any uppercase letter.
[A-Za-z] : matches any upper or lower case letter.
[A-Za-z0-9] : matches any upper or lower case letter or digit.
. : 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.
**** : escape the following special character
^ : only match if they are the first characters of a line
$: only match if they are the last characters of a line.
\b : only match at a word boundary.
***** matches the preceding element zero or more times.
+ matches the preceding element one or more times.
? matches when the preceding character appears zero or one time.
{VALUE} matches the preceding character the number of times defined by VALUE;
| : matches either/or.
/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.7

What will the regular expression

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

match?

Cheat Sheet

[ABC] : matches A or B or C.
[A-Z] : matches any uppercase letter.
[A-Za-z] : matches any upper or lower case letter.
[A-Za-z0-9] : matches any upper or lower case letter or digit.
. : 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.
**** : escape the following special character
^ : only match if they are the first characters of a line
$: only match if they are the last characters of a line.
\b : only match at a word boundary.
***** matches the preceding element zero or more times.
+ matches the preceding element one or more times.
? matches when the preceding character appears zero or one time.
{VALUE} matches the preceding character the number of times defined by VALUE;
| : matches either/or.
/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.7

What will the regular expression

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

match?

Either: “\b[Oo]rgani.e\b”

Or: “\b[Oo]rgani.e\w{1}\b”

Cheat Sheet

[ABC] : matches A or B or C.
[A-Z] : matches any uppercase letter.
[A-Za-z] : matches any upper or lower case letter.
[A-Za-z0-9] : matches any upper or lower case letter or digit.
. : 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.
**** : escape the following special character
^ : only match if they are the first characters of a line
$: only match if they are the last characters of a line.
\b : only match at a word boundary.
***** matches the preceding element zero or more times.
+ matches the preceding element one or more times.
? matches when the preceding character appears zero or one time.
{VALUE} matches the preceding character the number of times defined by VALUE;
| : matches either/or.
/i : case-insensitive (equivalent to `[A-Za-z]`).

Exercise 1.7

What will the regular expression

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

match?

Either: “\b[Oo]rgani.e\b”

Or: “\b[Oo]rgani.e\w{1}\b”

✔
organize
Organiser
organized
She organised

✘
Organisers
She organised
Organiseddd

Cheat Sheet

[ABC] : matches A or B or C.
[A-Z] : matches any uppercase letter.
[A-Za-z] : matches any upper or lower case letter.
[A-Za-z0-9] : matches any upper or lower case letter or digit.
. : 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.
**** : escape the following special character
^ : only match if they are the first characters of a line
$: only match if they are the last characters of a line.
\b : only match at a word boundary.
***** matches the preceding element zero or more times.
+ matches the preceding element one or more times.
? matches when the preceding character appears zero or one time.
{VALUE} matches the preceding character the number of times defined by VALUE;
| : matches either/or.
/i : case-insensitive (equivalent to `[A-Za-z]`).

Online tools to build and check regular expressions

To check your logic and see what strings your regular expression will match:

- **Regex101**: https://regex101.com/
- **RegexPal**: https://www.regexpal.com/
- **Myregexp**: https://myregexp.com/ (Javascript implementation)

Visualise the workflow of a regular expression:

- **Regexper**: https://regexper.com/

Exercise 2

- Work in your Breakout Room on the following exercises (5 mins)
- Syntax does not need to be perfect - more interested in the concept/approach for now
- More than one solution may be possible (some may be more concise/elegant than others!)
- We will return to the main room to share answers

Exercise 2

1. What will the regular expression **Fr[ea]nc[eh]** match?

2. What will the regular expression **Fr[ea]nc[eh]$** match?

3. What regular expression would match the strings “French” and “France” that appear at the beginning of a line?

4. How could you match the whole words colour and color (case insensitive)?

Solutions for Exercise 2

1. Fr[ea]nc[eh]

```
French
France
Frence
Franch
```

```
French-fried
```

```
In France
```

2. Fr[ea]nc[eh]$

```
French
France
Frence
Franch
```

```
In French
```

```
faux-French
```

3. Match the strings French and France that appear at the beginning of a line?

```
^France|^French
```

4. Match the whole words colour and color (case insensitive)?

```
Worked example
on next slide!
```

Possible matches?

Color COLOR color Colour COLOUR colour

use these in the "test string" pane of regex101.com

\b[Cc]olo[A-Za-z]r\b

Color COLOR color Colour COLOUR colour

\b[Cc]olou?r\b

Color COLOR color Colour COLOUR colour

\b[Cc]olou?r\b|\bCOLOU?R\b

Color COLOR color Colour COLOUR colour

or

\b[Cc]olou?r\b/i

Color COLOR color Colour COLOUR colour

/i flag renders the expression case insensitive

* on regex101.com, this expression would be written as \b[Cc]olou?r\b/gmi:
/g "global" flag searches through the text for all valid matches; /m "multiline" flag continues searching at end of the line

Exercise 3

1. How would you find the whole word headrest and or head rest but not head rest (that is, with two spaces between head and rest?

2. How would you find a string that ends with four letters preceded by at least one zero?

3. How do you match any four-digit string anywhere?

4. How would you match the date format dd-MM-yyyy?

Bonus Question:

How would you match publication formats such as British Library : London, 2015 and Manchester University Press: Manchester, 1999?

Solutions for Exercise 3

1. **\bhead ?rest\b** or **\bhead\s?rest\b**

(\s also matches other whitespaces e.g. tabs, newline characters - potential for false positives)

2. **0+[A-Za-z]{4}\b**

3. **\d{4}**
(Without word boundaries, you will also get matches to numbers with 5 or more digits: \b\d{4}\b is a more specific solution

4. **\b\d{2}-\d{2}-\d{4}\b**

(The word boundaries "\b" could be removed if your data is already formatted and cleaned)

Bonus Question:

A good starting solution is .* ?: .*, \d{4}

(Without word boundaries you will find that this matches any text you put before British or Manchester. Nevertheless, the regular expression does a good job on the first look up and may be need to be refined on a second, depending on your data.)

Schedule

~~10:00-10:15 Intro~~

~~10:15 -11:00. Introduction to Regular expressions 1~~

~~11:00-11:10 Coffee break 1~~ ☕

~~11:10-12:00 Regular expressions 2~~

12:00-13:00 Lunch break 🍽️

13:00-13:50 Matching & Extracting Strings 1

13:50-14:00 Coffee break 2 ☕

14:10-15:00 Matching & Extracting Strings 2

Schedule

~~10:00-10:15 Intro~~

~~10:15 -11:00. Introduction to Regular expressions 1~~

~~11:00-11:10 Coffee break 1~~

~~11:10-12:00 Introduction to Regular expressions 2~~

~~12:00-13:00 Lunch break~~

13:00-14:00 Matching & Extracting Strings 1

14:00-14:10 Coffee break 2

14:10-15:00 Matching & Extracting Strings 2

Don’t forget to sign back in to the attendance list for the evening (line 144)!

Matching & Extracting Strings

Before lunch, we learned about regular expressions and how to use them to build searches

Regular expressions can be used to:

- match words, email addresses, and phone numbers.
- extract substrings from strings (e.g. addresses).

Quick revision:
https://librarycarpentry.org/lc-data-intro/instructor/03-quiz.html

Exercise 4 - Live-coding

1. Open a browser and go to https://regex101.com
2. Open the swcCoC.md file (https://github.com/LibraryCarpentry/lc-data-intro/tree/main/episodes/data/swcCoC.md)

https://librarycarpentry.org/lc-data-intro/02-match-extract-strings.html#exercise-using-regex101.com

Schedule

~~10:00-10:15 Intro~~

~~10:15 -11:00. Introduction to Regular expressions 1~~

~~11:00-11:10 Coffee break 1~~ ☕

~~11:10-12:00 Introduction to Regular expressions 2~~

~~12:00-13:00 Lunch break~~ 🍽️

~~13:00-14:00 Matching & Extracting Strings 1~~

~~14:00-14:10 Coffee break 2~~ ☕

14:10-15:00 Matching & Extracting Strings 2 🎉

Exercise 5 - finding email addresses using regex101.com

Task: using the same file as before (swcCoC.md file), build a regular expression to find email addresses within the Code of Conduct.

Pointers:

- All email addresses contain which character?
- What types of characters can come before this symbol?
  - Pay attention to special characters here!
- What string patterns can come after
  - https://data.iana.org/TLD/tlds-alpha-by-domain.txt

https://librarycarpentry.org/lc-data-intro/02-match-extract-strings.html#exercise-finding-email-addresses-using-regex101.com

Exercise 6 - finding phone numbers, Using regex101.com

- Task: using the same file as before (swcCoC.md file), build a regular expression to find phone numbers within the Code of Conduct.

- Points to consider:
  - It may or may not have a country code, perhaps starting with a "+".
  - It will have an area code, potentially enclosed in parentheses.
  - It may have the sections all separated with a "-".

(530)341-3230

1-530-341-3230

+1 (530) 341-3230

country code | area code | line number

https://librarycarpentry.org/lc-data-intro/02-match-extract-strings.html#exercise-finding-phone-numbers-using-regex101.com

Exercise 7 - Google Sheets Demo

Target Format: 34020 NORTH FORK ROAD (-151.825607, 59.77965)

Latitude and longitude values:

- may be negative: optional "-" sign: -?
- Any number of digits: \d+ (repeats one or more times)
- Decimal point - full stop, need to escape (special character): \.
- Any number of digits after the decimal point: again, \d+

```
=REGEXEXTRACT(G2,"-?\d+\.\d+, -?\d+\.\d+")
```

Alternatively: just match between parentheses:

```
=REGEXEXTRACT(G2,"\(.*\)")
```

Closing out Day 3

- RegEx: Combining literal characters with metacharacters, allowing you to perform sophisticated pattern-matching
- Additional resources: links in the EtherPad
- It is ok not to do everything at once - do the easy things first.
- EtherPad questions:
  - How can you see yourself using Regular Expressions in your own work?
  - One thing you liked about today
  - One thing you would have changed about today?

Additional Resources

- Post-course embedding knowledge:
  - https://librarycarpentry.org/lc-data-intro/instructor/03-quiz.html
  - https://librarycarpentry.org/lc-data-intro/instructor/04-exercises.html
- Comparison of different RegEx implementations: https://gist.github.com/CMCDragonkai/6c933f4a7d713ef712145c5eb94a1816
- RegEx in R: https://bookdown.org/csgillespie/efficientR/data-carpentry.html#ref-sanchez_handling_2013
- Regular Expression Cheatsheet: **https://librarycarpentry.org/lc-data-intro/reference**
- Regex hexagonal puzzle: https://rampion.github.io/RegHex
- Regexper: https://regexper.com/
- Regexr: https://regexr.com/
- Regex 101: https://regex101.com/

Additional Resources

Question on the difference between \b (word boundary) and \B (non-word boundary)

For example if the string is `"Hello, world!"` then `\b` matches in the following places:

```
 H e l l o ,   w o r l d !
^         ^   ^         ^
```

And `\B` matches those places where `\b` doesn't match:

```
 H e l l o ,   w o r l d !
  ^ ^ ^ ^   ^   ^ ^ ^ ^   ^
```

https://stackoverflow.com/questions/4541573/what-are-non-word-boundary-in-regex-b-compared-to-word-boundary

Additional Resources: RegEx mindmap

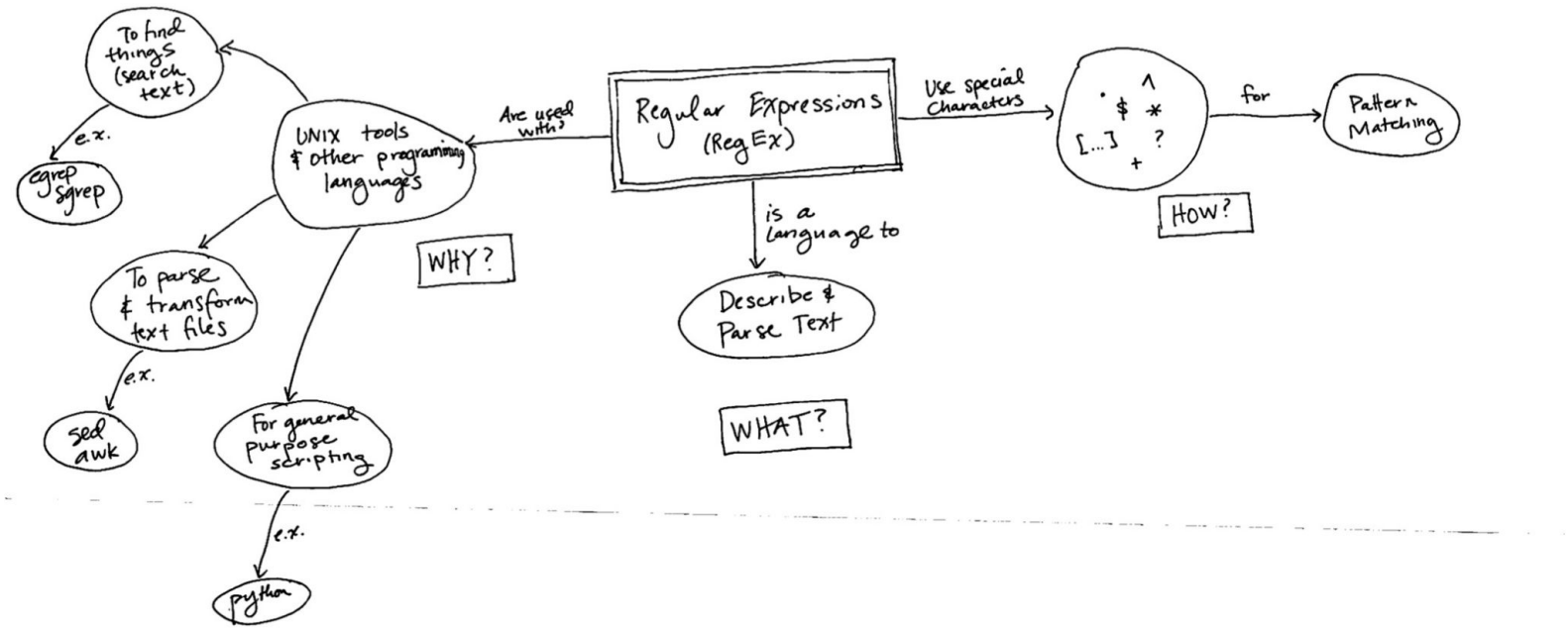


Image credit: Molly Gibson (https://swcarpentry.github.io/training-course/2013/08/concept-map-regular-expressions/)

Additional Resources: Comic Relief

https://xkcd.com/208/