Regular Expressions

1 / 16

What are they?

What do they do?

- Match on types of character (e.g. 'upper case letters', 'digits', 'spaces', etc.)
- Match patterns that repeat any number of times
- Capture the parts of the original string that match your pattern

The regular expression

```
organi[sz]e
```

matches both "organise" and "organize"

Brackets

- `[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 (note: this is case-sensitive)

Then there are:

- `.` 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
- `\` NB: this is also used to escape the following character when that character is a special character. So, for example, a regular expression that found `.com` would be `\.com` because `.` is a special character that matches any character.

And

- `^` asserts the position at the start of the line. So what you put after it will only match the first characters of a line or contents of a cell.
- `$` asserts the position at the end of the line. So what you put after it will only match the last character of a line of contents of a cell.
- `\b` adds a word boundary. Putting this either side of a stops the regular expression matching longer variants of words. So:
    - the regular expression `foobar` will match `foobar` and find `666foobar`, `foobar777`, `8thfoobar8th` et cetera
    - the regular expression `\bfoobar` will match `foobar` and find `foobar777`
    - the regular expression `foobar\b` will match `foobar` and find `666foobar`
    - the regular expression `\bfoobar\b` will find `foobar`

So, what is `^[Oo]rgani.e\b` going to match?

Other useful special characters are:

- `*` matches when the preceding character appears any number of times including zero
- `+` matches when the preceding character appears any number of times excluding zero
- `?` matches when the preceding character appears one or zero times
- `{VALUE}` matches the preceding character the number of times define by VALUE; ranges can be specified with the syntax `{VALUE,VALUE}`
- `|` means or.

So, what are these going to match?

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

```
[Oo]rgani.e\w+$
```

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

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

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

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