Advanced Regular Expressions

Syntax

CAPTURE GROUPS

- Extracting text using a capture group:

```
s.str.extract(pattern_with_capture_group)
```

- Extracting text using multiple capture groups:

```
s.str.extract(pattern_with_multiple_capture_groups)
```

SUBSTITUTION

- Substituting a regex match:

```
s.str.replace(pattern, replacement_text)
```

Concepts

- Capture groups allow us to specify one or more groups within our match that we can access separately.

Pattern	Explanation
`(yes)no`	Matches `yesno` , capturing `yes` in a single capture group.
`(yes)(no)`	Matches `yesno` , capturing `yes` and `no` in two capture groups.

- Backreferences allow us to repeat a capture group within our regex pattern by referring to them with an integer in the order they are captured.

Pattern	Explanation
`(yes)no\1`	Matches `yesnoyes`
`(yes)(no)\2\1`	Matches `yesnonoyes`

- Lookarounds let us define a positive or negative match before or after our string.

Pattern	**Explanation**
`zzz(?=abc)`	Matches `zzz` only when it is followed by `abc`
`zzz(?!abc)`	Matches `zzz` only when it is not followed by `abc`
`(?<=abc)zzz`	Matches `zzz` only when it is preceded by `abc`
`(?<!zzz)abc`	Matches `zzz` only when it is not preceded by `abc`

Resources

- re module
- RegExr Regular Expression Builder