Wrangling data with Regular Expressions

Open these in a browser:

regex101.com

bit.ly/nicar-regex

bit.ly/nicar-regex-sample

bit.ly/nicar-regex-cheats

Who am I?

Christian McDonald
University of Texas at Austin
School of Journalism and Media

christian.mcdonald@utexas.edu

Regular expression patterns

A quick look at a key concept

Let's say, you have a column of data with a bunch of this:

512-555-1212

But you want this:

(512) 555-1212

You could replace this

512-555-1212

With this

(512) 555-1212

But what if you start with this?

512-555-1212

301-555-1213

404-555-1212

Instead of searching for the exact text

512-555-1212

Regex lets you search for types of things

```
^(\d{3})-555-1212
```

Which translates to “starting at the beginning of a line, find three numbers together and capture them for later, then a dash.”

Since we have captured text in a group, we can replace it as a group

($1) 555-1212

For each line

($1) 555-1212

($1) 555-1213

($1) 555-1212

So you end up with

(512) 555-1212

(301) 555-1213

(404) 555-1212

Regex uses text as commands, which we call tokens

- `^` finds the beginning of a line
- `*` will find "zero or more" of whatever precedes it

\ is special. It turns letters into tokens

- `\d` will find any number character (or digit).
- `\D` will match anything other than a number.
- `\t` is a tab character.

Since we are using text both as characters and tokens, we use \ to escape between their action and their literal meaning

- `\*` to get an actual asterisk, because `*` by itself means "find zero or more".

It might be easier to show than explain

bit.ly/nicar-regex