Regular expressions

Warmup

Work on the activity (handout), then we will discuss as a class.

Warmup

- computer professionals celebrate 10th birthday of a.l.i.c.e.
- i watched ‘home alone’ for the first time and it was actually horrifying
- f.b.i. lab houses growing database of dna profiles
- 6 things i wish i knew as a teen
- i asked my mom for marriage advice and here’s what happened
- i wore food on my face instead of makeup to see if anyone would notice

Of these 6 headlines, 4 are clickbait. All the clickbait headlines are written in first person. How can I detect these headlines?

Identifying first person headlines

What’s wrong with this code?

```r
str_subset(headlines, "i")
```

```
[1] "i watched \"home alone\" for the first time and it was actually
horrifying"
[2] "computer professionals celebrate 10th birthday of a.l.i.c.e."
[3] "f.b.i. lab houses growing database of dna profiles"
[4] "6 things i wish i knew as a teen"
[5] "i asked my mom for marriage advice and here's what happened"
[6] "i wore food on my face instead of makeup to see if anyone would
notice"
```

Identifying first person headlines

Adding word boundaries:

\\b : special character for a word boundary

```r
str_subset(headlines, "\\bi\\b")
```

```
[1] "i watched \"home alone\" for the first time and it was actually
horrifying"
[2] "computer professionals celebrate 10th birthday of a.l.i.c.e."
[3] "f.b.i. lab houses growing database of dna profiles"
[4] "6 things i wish i knew as a teen"
[5] "i asked my mom for marriage advice and here's what happened"
[6] "i wore food on my face instead of makeup to see if anyone would
notice"
```

word boundaries!

How else could we modify this pattern?

Identifying first person headlines

The word "I" is likely to either *start* the headline, or be preceded by a *space*:

^ : Start of string

\\s : a space

| : alternation

```r
str_subset(headlines, "(^|\\s)i\\b")
```

```
[1] "i watched \"home alone\" for the first time and it was actually
horrifying"
[2] "6 things i wish i knew as a teen"
[3] "i asked my mom for marriage advice and here's what happened"
[4] "i wore food on my face instead of makeup to see if anyone would
notice"
```

where this fails:

Acronym:

I.R.S,

Allow for different cases:

(^|\\s)(i|I)\\b

Avoid things like I.R.S. :

(^|\\s)i(,|\\s|\\))

Regular expressions so far

Regular expression: a tool for specifying a search pattern in text.

Some regular expressions so far:

- `\d` any digit
- `+` one or more occurrences
- `^` anchors at the beginning
- `$` anchors at the end
- `\b` word boundary
- `|` alternation (this pattern OR that pattern)

Example 2: Cleaning phone numbers

You are working with customer data in which customers have entered their phone numbers:

```
[1] "555 867-5309"   "555 123 1234"   "(555) 298-9090" "(555) 095 9876"
[5] "5553246789"
```

You want to clean the numbers so they all have the same form. What would be the easiest approach?

Example 2: Cleaning phone numbers

```r
str_remove_all(phone_nums, "\\D")
```

```
[1] "5558675309" "5551231234" "5552989090" "5550959876" "5553246789"
```

- `str_remove_all` removes all matches to a pattern
- `\d` matches any digit
- `\D` matches any *non*-digit

Shorthand character classes

- `\d` matches any digit
- `\w` matches any “word character” (letters, digits, underline)
- `\s` matches any “whitespace character” (space, tab, enter, new line)
- `\D`, `\W`, and `\S` are negations of `\d`, `\w`, and `\s`

Example 3: Selecting files

Here are a set of files that live on my computer:

```r
file_names
```

```
[1] "research/project1/code.R"
[2] "research/project1/data.csv"
[3] "research/project2/sim_output.csv"
[4] "teaching/sta279/lecture1.qmd"
[5] "teaching/sta279/example_data.csv"
[6] "teaching/sta279/research_project.html"
```

How would I select only the files that live in the *research* folder?

Example 3: Selecting files

Here are a set of files that live on my computer:

```r
file_names
```

```
[1] "research/project1/code.R"
[2] "research/project1/data.csv"
[3] "research/project2/sim_output.csv"
[4] "teaching/sta279/lecture1.qmd"
[5] "teaching/sta279/example_data.csv"
[6] "teaching/sta279/research_project.html"
```

How would I select only the files that live in the *research* folder?

```r
str_subset(file_names, "^research")
```

```
[1] "research/project1/code.R"         "research/project1/data.csv"
[3] "research/project2/sim_output.csv"
```

Example 3: Selecting files

Here are a set of files that live on my computer:

```r
file_names
```

```
[1] "research/project1/code.R"
[2] "research/project1/data.csv"
[3] "research/project2/sim_output.csv"
[4] "teaching/sta279/lecture1.qmd"
[5] "teaching/sta279/example_data.csv"
[6] "teaching/sta279/research_project.html"
```

What about only *csv* files in the research folder?

Start with research

(then have something else)

end with .csv

Example 3: Selecting files

Here are a set of files that live on my computer:

```r
file_names
```

```
[1] "research/project1/code.R"
[2] "research/project1/data.csv"
[3] "research/project2/sim_output.csv"
[4] "teaching/sta279/lecture1.qmd"
[5] "teaching/sta279/example_data.csv"
[6] "teaching/sta279/research_project.html"
```

What about only *csv* files in the research folder?

```r
str_subset(file_names, "^research.+\\.csv$")
```

```
[1] "research/project1/data.csv"
"research/project2/sim_output.csv"
```

match anything

.csv

. special character that matches any character

=> .+ (any string with one or more characters)

\\. literal period

Dot

The `.` is a special character which matches (almost) any character:

```r
str_view("I like bananas.", ".")
```

```
[1] │ <I>< ><l><i><k><e>< ><b><a><n><a><n><a><s><.>
```

If we want to match a *literal* period, we need to escape it:

```r
str_view("I like bananas.", "\\.")
```

```
[1] │ I like bananas<.>
```

Example 4: Extracting LaTeX

LaTeX is a tool for scientific and mathematical typesetting. For example:

```
$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$
```

becomes

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Example 4: Extracting LaTeX

Suppose we have a document which contains equations in LaTeX:

```
document_text
```

```
[1] "The equation for the simple linear regression line is given by
$Y_i = \\beta_0 + \\beta_1 X_i + \\varepsilon_i$"
```

Question: If I want to extract only the equation, what pattern am I trying to match?

Example 4: Extracting LaTeX

```r
document_text
```

```
[1] "The equation for the simple linear regression line is given by
$Y_i = \\beta_0 + \\beta_1 X_i + \\varepsilon_i$"
```

```r
str_extract(document_text, "\\$.+\\$")
```

```
[1] "$Y_i = \\beta_0 + \\beta_1 X_i + \\varepsilon_i$"
```

- Remember that `$` is a special character in regular expressions, meaning “the end of the string”. To get a *literal* dollar sign, we need the escape character: `\\$`

Example 4: Extracting LaTeX

```r
document_text
```

```
[1] "The equation for the simple linear regression line is given by
$Y_i = \\beta_0 + \\beta_1 X_i + \\varepsilon_i$"
```

```r
str_extract(document_text, "(?<=\\$).+(?=\\$)")
```

```
[1] "Y_i = \\beta_0 + \\beta_1 X_i + \\varepsilon_i"
```

come after $ come before another $

- `(?<=)` is a *positive lookbehind*. It is used to identify expressions which are *preceded* by a particular expression
- `(?=)` is a *positive lookahead*. It is used to identify expressions which are *followed* by a particular expression

Class activity

- Work independently or with a neighbor on the class activity
- At the end of class, submit your work as an HTML file on Canvas (one per group, list all your names)

For next time, read:

- Chapter 15.4 - 15.7 in *R for Data Science*