Regular expression basics

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Kasey Jones
Research Data Scientist

What is natural language processing?

NLP:

- Focuses on using computers to analyze and understand text

Topics Covered:

- Classifying Text
- Topic Modeling
- Named Entity Recognition
- Sentiment Analysis

What are regular expressions?

- A sequence of characters used to search text
- Examples include:
  - searching files in a directory using the command line
  - finding articles that contain a specific pattern
  - replacing specific text
  - ...

Examples

```
words <- c("DW-40", "Mike's Oil", "5w30", "Joe's Gas", "Unleaded", "Plus-89")
```

```
# Finding Digits
grep("\\d", words)
```

```
[1] 1 3 6
```

```
# Finding Apostrophes
grep("\\'", words)
```

```
[1] "Mike's Oil"     "Joe's Gasoline"
```

Regular Expression Examples

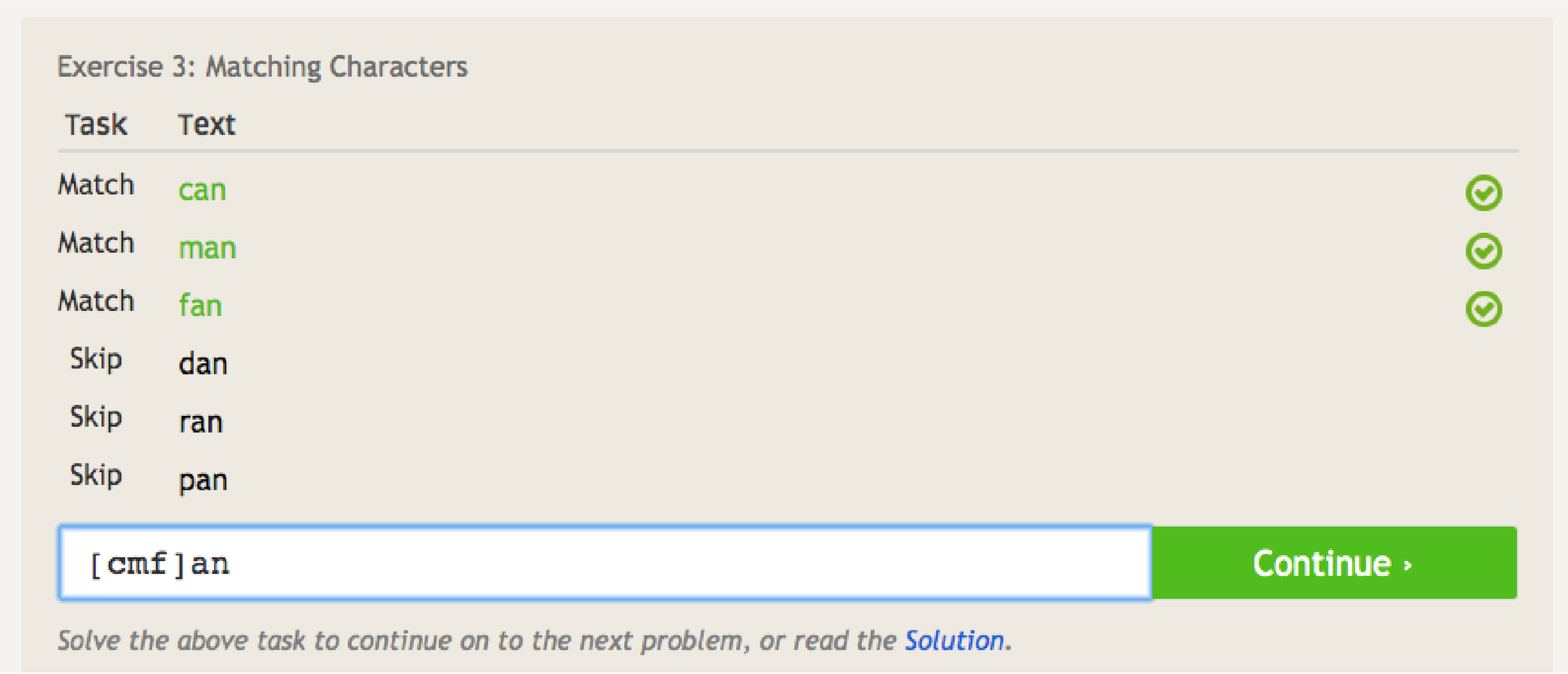
Pattern	Text Matches	R Example	Text Example
\w	Any alphanumeric	gregexpr(pattern ='\w', <text>)	a
\d	Any digit	gregexpr(pattern ='\d', text)	1
\w+	An alphanumeric of any length	gregexpr(pattern ='\w+', text)	word
\d+	Digits of any length	gregexpr(pattern ='\d+', text)	1234
\s	Spaces	gregexpr(pattern ='\s', text)	' '
\S	Any non-space	gregexpr(pattern ='\S', text)	word

R Examples

Function	Purpose	Syntax
grep	Find matches of the pattern in a vector	grep(pattern ='\w', x = <vector>, value = F)
gsub	Replaces all matches of a string/vector	gsub(pattern ='\d+', replacement = "", x = <vector>)

RegEx Practice

Regular Expression Practice



[1] https://regexone.com/lesson/matching_characters

Time to code!

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Tokenization

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Kasey Jones
Research Data Scientist

What are tokens?

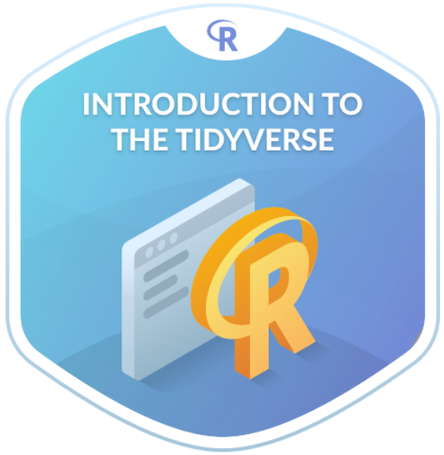
Common types of tokenization:

- characters
- words
- sentences
- documents
- regular expression separations

tidytext package

Package overview:

- "Text Mining using `dplyr` , `ggplot2` , and Other Tidy Tools"
- Follows the tidy data format



[1] https://cran.r [2] project.org/web/packages/tidytext/index.html

The Animal Farm dataset

```
animal_farm
```

```
# A tibble: 10 x 2
   chapter    text_column
   <chr>      <chr>
 1 Chapter 1  "Mr. Jones, of the Manor Farm, had locked ...
 2 Chapter 2  "Three nights later old Major died peacefully ...
 3 Chapter 3  "How they toiled and sweated to get the hay ...
...
```

[1] https://en.wikipedia.org/wiki/Animal_Farm

Tokenization practice

```
animal_farm %>%
  unnest_tokens(output = "word",
                input = text_column,
                token = "words")
```

Token Options

- `sentences`
- `lines`
- `regex`
- `words`
- …

Counting tokens

```
animal_farm %>%
  unnest_tokens(output = "word",
                token = "words",
                input = text_column) %>%
  count(word, sort = TRUE)
```

```
# A tibble: 4,076 x 2
   word      n
   <chr> <int>
 1 the    2187
 2 and     966
 3 of      899
 4 to      814
 ...
```

Tokenization with regular expressions

```
animal_farm %>%
  filter(chapter == 'Chapter 1') %>%
  unnest_tokens(output = "Boxer", input = text_column,
                token = "regex", pattern = "(?i)boxer") %>%
  slice(2:n())
```

```
# A tibble: 5 x 2
  chapter   Boxer
  <chr>     <chr>
2 Chapter 1 " and clover, came in together, walking very slowly and setting down their vast hairy ho
3 Chapter 1 " was an enormous beast, nearly eighteen hands high, and as strong as any two ordinary h
4 Chapter 1 "; the two of them usually spent their sundays together in the small paddock beyond the
...
```

Let's tokenize some text.

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Text cleaning basics

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Kasey Jones
Research Data Scientist

The Russian tweet data set

3 Million Russian Troll Tweets

B	C	D	E	F	G	H	I	J
external_author_id	author	content	region	language	publish_date	harvested_date	following	followers
9.06E+17	10_GOP	"We have a s	Unknown	English	10/1/17 19:58	10/1/17 19:59	1052	9636
9.06E+17	10_GOP	Marshawn Ly	Unknown	English	10/1/17 22:43	10/1/17 22:43	1054	9637
9.06E+17	10_GOP	Daughter of	Unknown	English	10/1/17 22:50	10/1/17 22:51	1054	9637
9.06E+17	10_GOP	JUST IN: Pres	Unknown	English	10/1/17 23:52	10/1/17 23:52	1062	9642
9.06E+17	10_GOP	19,000 RESP	Unknown	English	10/1/17 2:13	10/1/17 2:13	1050	9645
9.06E+17	10_GOP	Dan Bongino	Unknown	English	10/1/17 2:47	10/1/17 2:47	1050	9644

- We will explore the first 20,000 tweets
- Data includes the tweet, followers, following, publish date, account type, etc.
- Great dataset for topic modeling, classification, named entity recognition, etc.

[1] https://github.com/fivethirtyeight/russian [2] troll [3] tweets

Top occurring words

```
library(tidytext); library(dplyr)
russian_tweets %>%
  unnest_tokens(word, content) %>%
  count(word, sort = TRUE)
```

```
# A tibble: 44,318 x 2
   word      n
   <chr> <int>
 1 t.co  18121
 2 https 16003
 3 the    7226
 4 to     5279
 ...
```

Remove stop words

```
tidy_tweets <- russian_tweets %>%
  unnest_tokens(word, content) %>%
  anti_join(stop_words)
```

```
tidy_tweets %>%
  count(word, sort = TRUE)
```

```
 1 t.co              18121
 2 https             16003
 3 http               2135
 4 blacklivesmatter   1292
 5 trump              1004
 ...
```

```
# A tibble: 1,149 x 2
   word        lexicon
   <chr>       <chr>
 1 a           SMART
 2 a's         SMART
 3 able        SMART
 4 about       SMART
 5 above       SMART
```

Custom stop words

```
custom <- add_row(stop_words, word = "https", lexicon = "custom")
custom <- add_row(custom, word = "http", lexicon = "custom")
custom <- add_row(custom, word = "t.co", lexicon = "custom")
```

```
russian_tweets %>%
  unnest_tokens(word, content) %>%
  anti_join(custom) %>%
  count(word, sort = TRUE)
```

Final results

```
# A tibble: 43,663 x 2
   word                n
   <chr>           <int>
 1 blacklivesmatter  1292
 2 trump             1004
 3 black              781
 4 enlist             764
 5 police             745
 6 people             723
 7 cops               693
```

Stemming

- ***enlist***ed ---> enlist
- ***enlist***ing ---> enlist

```
library(SnowballC)
tidy_tweets <- russian_tweets %>%
  unnest_tokens(word, content) %>%
  anti_join(custom)
# Stemming
stemmed_tweets <- tidy_tweets %>%
  mutate(word = wordStem(word))
```

Stemming Results

```
# A tibble: 38,907 x 2
   word              n
   <chr>         <int>
 1 blacklivesmatt  1301
 2 cop             1016
 3 trump           1013
 4 black            848
 5 enlist           809
 6 polic            763
 7 peopl            730
```

Example time.

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R