Text analysis and regular expressions

Nicholas Horton (nhorton@amherst.edu)

May 19, 2022

Note: you'll need to:

1. install the DickinsonPoems package and
2. run `get_sentiments("afinn")` interactively

to be able to knit this.

The text analytics chapter in MDSR2 has more examples and exercises: https://mdsr-book.github.io/mdsr2e/ch-text.html

Regular expressions

Regular expressions are important in text processing. Wikipedia describe them as follows:

- Each character in a regular expression is either understood to be a metacharacter with its special meaning, or a regular character with its literal meaning.
- Together, they can be used to identify textual material of a given pattern, or process a number of instances of it that can vary from a precise equality to a very general similarity of the pattern.
- The pattern sequence itself is an expression that is a statement in a language designed specifically to represent prescribed targets in the most concise and flexible way to direct the automation of text processing of general text files, specific textual forms, or of random input strings.

Warmup #1

I provide a workshop to students with the output suppressed and have them work in groups to explain what is happening for each of the 23 patterns being tested.

They are asked to put their summaries on the board, then start to work on the next problems

```
x <- c("popular", "popularity", "popularize", "popularise",
       "Popular", "Population", "repopulate", "reproduce",
       "happy family", "happier\tfamily", " happy family", "P6dn")
grep(pattern = 'pop', x)  #1
```

```
## [1] 1 2 3 4 7
```

```
grep(pattern = '^pop', x) #2
```

```
## [1] 1 2 3 4
```

```
grep(pattern = 'populari[sz]e', x) #3
```

```
## [1] 3 4
```

```
grep(pattern = 'pop.*e', x) #4
```

```
## [1] 3 4 7
```

```
grep(pattern = 'p[a-z]*e', x) #5
```

```
## [1]  3  4  7  8 10
```

```
grep(pattern = '^[Pp][a-z]+.*n', x) #6
```

```
## [1] 6
```

```
grep(pattern = '^[^Pp]', x) #7
```

```
## [1]  7  8  9 10 11
```

```
grep(pattern = '^[A-Za-p]', x) #8
```

```
## [1]  1  2  3  4  5  6  9 10 12
```

```
grep(pattern = '[ ]', x) #9
```

```
## [1]  9 11
```

```
grep(pattern = '[\t]', x) #10
```

```
## [1] 10
```

```
grep(pattern = '[ \t]', x) #11
```

```
## [1]  9 10 11
```

```
grep(pattern = '^[ ]', x) #12
```

```
## [1] 11
```

```
nchar(x)==7 #13
```

```
##  [1]  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
ds <- DickinsonPoems::get_poem("gutenberg3.txt249")
ds <- ds[ds != " "]
ds
```

```
##  [1] "XIII. "
##  [2] "PARTING. "
##  [3] "My life closed twice before its close; "
##  [4] "  It yet remains to see "
##  [5] "If Immortality unveil "
##  [6] "  A third event to me, "
##  [7] "So huge, so hopeless to conceive, "
##  [8] "  As these that twice befell. "
##  [9] "Parting is all we know of heaven, "
## [10] "  And all we need of hell. "
```

```
grep('third|twice', ds)   #14
```

```
## [1] 3 6 8
```

```
grep('\\.', ds)  #15
```

```
## [1]  1  2  8 10
```

```
grep('all|me|we', ds) #16
```

```
## [1]  6  9 10
```

```
options(scipen=5)
x <- c("$100.04", "100,000", "1000", "no more than $10,000")
as.numeric(x)  #17
```

```
## Warning: NAs introduced by coercion
```

```
## [1]   NA   NA 1000   NA
```

```
as.numeric(gsub("[$,.]", "", x)) #18
```

```
## Warning: NAs introduced by coercion
```

```
## [1]  10004 100000   1000     NA
```

```
readr::parse_number(x) #19
```

```
## [1]    100.04 100000.00   1000.00  10000.00
```

```
library(qdapRegex)
```

```
## Warning: package 'qdapRegex' was built under R version 4.1.2
```

```
##
## Attaching package: 'qdapRegex'
```

```
## The following object is masked from 'package:dplyr':
##
##     explain
```

```
## The following object is masked from 'package:ggplot2':
##
##     %+%
```

```
x <- c("Scene 1 (in a dark room)", "Where are the #tweeters?")
rm_round(x) #20
```

```
## [1] "Scene 1"                  "Where are the #tweeters?"
```

```
rm_round(x, extract = TRUE) #21
```

```
## [[1]]
## [1] "in a dark room"
##
## [[2]]
## [1] NA
```

```
unlist(rm_hash(x, extract = TRUE)) #22
```

```
## [1] NA          "#tweeters"
```

```
unlist(rm_between(ds, "so", "to", extract = TRUE))   #23
```

```
##  [1] NA          NA          NA          NA          NA          NA
##  [7] "hopeless" NA          NA          NA
```

Warmup #2

Next I start to have students work through other text data formats.

How often does the word "science" appear in Emily Dickinson's poetry?

```
all_poems <- list_poems() %>%
  map(get_poem) %>%
```

```
  unlist() %>%
  enframe(value = "words") %>%
  unnest_tokens(word, words)
head(all_poems)
```

```
## # A tibble: 6 x 2
##    name word
##   <int> <chr>
## 1     1 i
## 2     3 success
## 3     5 published
## 4     5 in
## 5     5 a
## 6     5 masque
```

```
all_poems %>%
  filter(stringr::str_detect(word, "science"))
```

```
## # A tibble: 4 x 2
##    name word
##   <int> <chr>
## 1   511 conscience
## 2  4618 science
## 3  6549 sciences
## 4  7849 science
```

Warmup #3

Finally, I introduce sentiment analysis.

Classify Emily Dickinson's poem *The Lonely House* as either positive or negative using the `AFINN` lexicon. Does this match with your own interpretation of the poem?

```
poem <- get_poem("gutenberg1.txt014")
poem
```

```
##  [1] "XV. "
##  [2] " "
##  [3] "THE LONELY HOUSE. "
##  [4] " "
##  [5] "I know some lonely houses off the road "
##  [6] "A robber 'd like the look of, -- "
##  [7] "Wooden barred, "
##  [8] "And windows hanging low, "
##  [9] "Inviting to "
## [10] "A portico, "
## [11] "Where two could creep: "
## [12] "One hand the tools, "
## [13] "The other peep "
## [14] "To make sure all's asleep. "
## [15] "Old-fashioned eyes, "
## [16] "Not easy to surprise! "
## [17] " "
## [18] "How orderly the kitchen 'd look by night, "
## [19] "With just a clock, -- "
## [20] "But they could gag the tick, "
```

```
## [21] "And mice won't bark; "
## [22] "And so the walls don't tell, "
## [23] "None will. "
## [24] " "
## [25] "A pair of spectacles ajar just stir -- "
## [26] "An almanac's aware. "
## [27] "Was it the mat winked, "
## [28] "Or a nervous star? "
## [29] "The moon slides down the stair "
## [30] "To see who's there. "
## [31] " "
## [32] "There's plunder, -- where? "
## [33] "Tankard, or spoon, "
## [34] "Earring, or stone, "
## [35] "A watch, some ancient brooch "
## [36] "To match the grandmamma, "
## [37] "Staid sleeping there. "
## [38] " "
## [39] "Day rattles, too, "
## [40] "Stealth's slow; "
## [41] "The sun has got as far "
## [42] "As the third sycamore. "
## [43] "Screams chanticleer, "
## [44] "\"Who's there?\" "
## [45] "And echoes, trains away, "
## [46] "Sneer -- \"Where?\" "
## [47] "While the old couple, just astir, "
## [48] "Fancy the sunrise left the door ajar! "
## [49] " "
## [50] " "
## [51] " "
## [52] " "
```

```
# need to first run `get_sentiments("afinn")` interactively.
poem_sentiments <- data.frame(poem) %>%
  unnest_tokens(word, poem) %>%
  inner_join(get_sentiments("afinn"))
```

```
## Joining, by = "word"
```

```
sum(poem_sentiments$value) # negative value
```

```
## [1] -8
```

Homework

Use the collection of Emily Dickinson poems to identify patterns or insights. This is a very open ended assignment: you may pick a few poems or consider all of them.