STAT 231: Problem Set 5A

Jack Dove

due by 5 PM on Monday, September 28

In order to most effectively digest the textbook chapter readings – and the new R commands each presents – series A homework assignments are designed to encourage you to read the textbook chapters actively and in line with the textbook's Prop Tip of page 33:

"**Pro Tip**: If you want to learn how to use a particular command, we highly recommend running the example code on your own"

A more thorough reading and light practice of the textbook chapter prior to class allows us to dive quicker and deeper into the topics and commands during class. Furthermore, learning a programming lanugage is like learning any other language – practice, practice, practice is the key to fluency. By having two assignments each week, I hope to encourage practice throughout the week. A little coding each day will take you a long way!

Series A assignments are intended to be completed individually. While most of our work in this class will be collaborative, it is important each individual completes the active readings. The problems should be straightforward based on the textbook readings, but if you have any questions, feel free to ask me!

Steps to proceed:

1. In RStudio, go to File > Open Project, navigate to the folder with the course-content repo, select the course-content project (course-content.Rproj), and click "Open"
2. Pull the course-content repo (e.g. using the blue-ish down arrow in the Git tab in upper right window)
3. Copy ps5A.Rmd from the course repo to your repo (see page 6 of the GitHub Classroom Guide for Stat231 if needed)
4. Close the course-content repo project in RStudio
5. Open YOUR repo project in RStudio
6. In the ps5A.Rmd file in YOUR repo, replace "YOUR NAME HERE" with your name
7. Add in your responses, committing and pushing to YOUR repo in appropriate places along the way
8. Run "Knit PDF"
9. Upload the pdf to Gradescope. Don't forget to select which of your pages are associated with each problem. *You will not get credit for work on unassigned pages (e.g., if you only selected the first page but your solution spans two pages, you would lose points for any part on the second page that the grader can't see).*

1. Text as Data

a.

In Section 19.1.1, the `grep` and `grepl` functions are introduced for detecting a pattern in a character vector (like finding a needle in a haystack). The following three calls look similar, but return different results. Explain what the 6 returned records indicate in each case:

- `head(grepl("  MACBETH", macbeth))`
- `head(grep("  MACBETH", macbeth, value = TRUE))`
- `head(grep("  MACBETH", macbeth))`

(Yes, the textbook explains the differences in these commands/calls to these commands, but it can be helpful if you run the lines yourself as well to be sure they work as you'd expect and to inspect the results.)

> ANSWER: First, the grep function returns six numbers, which represent the lines of MACBETH occurances in macbeth. Second, the grep function with the value definition returns the six exact strings with MACBETH occurances. Third, the grepl function returns a boolean of the precense of MACBETH in the first six lines of the entire play.

```r
# defining "macbeth" object
macbeth_url <- "http://www.gutenberg.org/cache/epub/1129/pg1129.txt"
Macbeth_raw <- RCurl::getURL(macbeth_url)
data(Macbeth_raw)
#Macbeth_raw

# strsplit returns a list: we only want the first element
macbeth <- stringr::str_split(Macbeth_raw, "\r\n")[[1]]
class(macbeth)
```

```
## [1] "character"
```

```r
length(macbeth)
```

```
## [1] 3194
```

```r
### finding literal strings
head(grep("  MACBETH", macbeth))
```

```
## [1] 228 433 443 466 478 483
```

```r
head(grep("  MACBETH", macbeth, value = TRUE))
```

```
## [1] "  MACBETH, Thane of Glamis and Cawdor, a general in the King's"
## [2] "  MACBETH. So foul and fair a day I have not seen."
## [3] "  MACBETH. Speak, if you can. What are you?"
## [4] "  MACBETH. Stay, you imperfect speakers, tell me more."
## [5] "  MACBETH. Into the air, and what seem'd corporal melted"
## [6] "  MACBETH. Your children shall be kings."
```

```r
head(grepl("  MACBETH", macbeth))
```

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

b.

Section 19.1.1 also introduces regular expressions. Why do the two lines below differ in their results?

- `head(grep("MACBETH\\.", macbeth, value = TRUE))`
- `head(grep("MACBETH.", macbeth, value = TRUE))`

(Yes, the textbook explains the differences, but it can be helpful if you run the lines yourself as well to be sure they work as you'd expect and to inspect the results.)

> ANSWER: The first function searched for only "MACBETH." Meanwhile, the second function uses a metacharacter, which allows for any successive character after MACBETH, which is why two of the six results display MACBETH before a comma.

```r
head(grep("MACBETH\\.", macbeth, value = TRUE))
```

```
## [1] "  MACBETH. So foul and fair a day I have not seen."
## [2] "  MACBETH. Speak, if you can. What are you?"
## [3] "  MACBETH. Stay, you imperfect speakers, tell me more."
## [4] "  MACBETH. Into the air, and what seem'd corporal melted"
## [5] "  MACBETH. Your children shall be kings."
## [6] "  MACBETH. And Thane of Cawdor too. Went it not so?"
```

```r
head(grep("MACBETH.", macbeth, value = TRUE))
```

```
## [1] "  MACBETH, Thane of Glamis and Cawdor, a general in the King's"
## [2] "  LADY MACBETH, his wife"
## [3] "  MACBETH. So foul and fair a day I have not seen."
## [4] "  MACBETH. Speak, if you can. What are you?"
## [5] "  MACBETH. Stay, you imperfect speakers, tell me more."
## [6] "  MACBETH. Into the air, and what seem'd corporal melted"
```

c.

The `stringr` package from the `tidyverse` collection of packages, has functions that work equivalently as `grep` and `grepl`. In particular:

- `str_detect(string=, pattern=)` is equivalent to `grepl(pattern=, x=)`
- `str_which(string=, pattern=)` is equivalent to `grep(pattern=, x=)`
- `str_subset(string=, pattern=)` is equivalent to `grep(pattern=, x=, value=TRUE)`

Uncomment and run the code below to ensure the same results are returned for each different pair (at least for the first six records). In words, explain what overall pattern is being searched for (i.e., what does the pattern "MAC[B-Z]" indicate?)?

> ANSWER: The pattern being searched for is any occurance of MAC with a capital letter B through Z (non-A) immediately following it. For example, MACINTOSH would fit the pattern, while MACAULAY would not.

```
head(grepl("MAC[B-Z]", macbeth))
```

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

```
head(str_detect(macbeth, "MAC[B-Z]"))
```

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

```
head(grep("MAC[B-Z]", macbeth))
```

```
## [1]  11 110 204 218 228 230
```

```
head(str_which(macbeth,"MAC[B-Z]"))
```

```
## [1]  11 110 204 218 228 230
```

```
head(grep("MAC[B-Z]", macbeth, value = TRUE))
```

```
## [1] "MACHINE READABLE COPIES MAY BE DISTRIBUTED SO LONG AS SUCH COPIES"
## [2] "MACHINE READABLE COPIES OF THIS ETEXT, SO LONG AS SUCH COPIES"
## [3] "WITH PERMISSION.  ELECTRONIC AND MACHINE READABLE COPIES MAY BE"
## [4] "THE TRAGEDY OF MACBETH"
## [5] "  MACBETH, Thane of Glamis and Cawdor, a general in the King's"
## [6] "  LADY MACBETH, his wife"
```

```
head(str_subset(macbeth, "MAC[B-Z]"))
```

```
## [1] "MACHINE READABLE COPIES MAY BE DISTRIBUTED SO LONG AS SUCH COPIES"
## [2] "MACHINE READABLE COPIES OF THIS ETEXT, SO LONG AS SUCH COPIES"
## [3] "WITH PERMISSION.  ELECTRONIC AND MACHINE READABLE COPIES MAY BE"
## [4] "THE TRAGEDY OF MACBETH"
## [5] "  MACBETH, Thane of Glamis and Cawdor, a general in the King's"
## [6] "  LADY MACBETH, his wife"
```

d. OPTIONAL

In section 19.2.2, the `wordcloud` package is used to create a word cloud based on text in abstracts from Data Science articles in arXiv (which is "a fast-growing electronic repository of preprints of scientific papers from many disciplines"). I've provided some code below to get you started coding along with the example. *This part (d) will not be graded, but is included to encourage you to test and explore some of the code in the extended example.* What words are included in `tidytexts`'s `stop_words` dataset? Do you think all of these words should be considered stop words (i.e. excluded from analysis) in all scenarios? Are there any that might be useful in some contexts?

> ANSWER: Stop words represent non-important words, like “and”, that are ignored in text analysis. I think stop words should be considered differently for each study. For example, in a study of boys’ and girls’ tweets to predict career path, “numbers” would be a great word to track, and yet is a stop word in the stop_words dataset.

```
# note that the tidytext, aRxiv, and wordcloud packages were loaded in the
# setup code chunk at the top of the program

# arxiv_search() is from the aRxiv package
DataSciencePapers <- arxiv_search(
  query = '"Data Science"',
  limit = 20000,
  batchsize = 100
)
```

```
## Total records matching query: 1147

## retrieved batch 1

## retrieved batch 2

## retrieved batch 3

## retrieved batch 4

## retrieved batch 5

## retrieved batch 6

## retrieved batch 7

## retrieved batch 8

## retrieved batch 9

## retrieved batch 10

## retrieved batch 11

## retrieved batch 12

## retrieved batch 13
```

```
glimpse(DataSciencePapers)
```

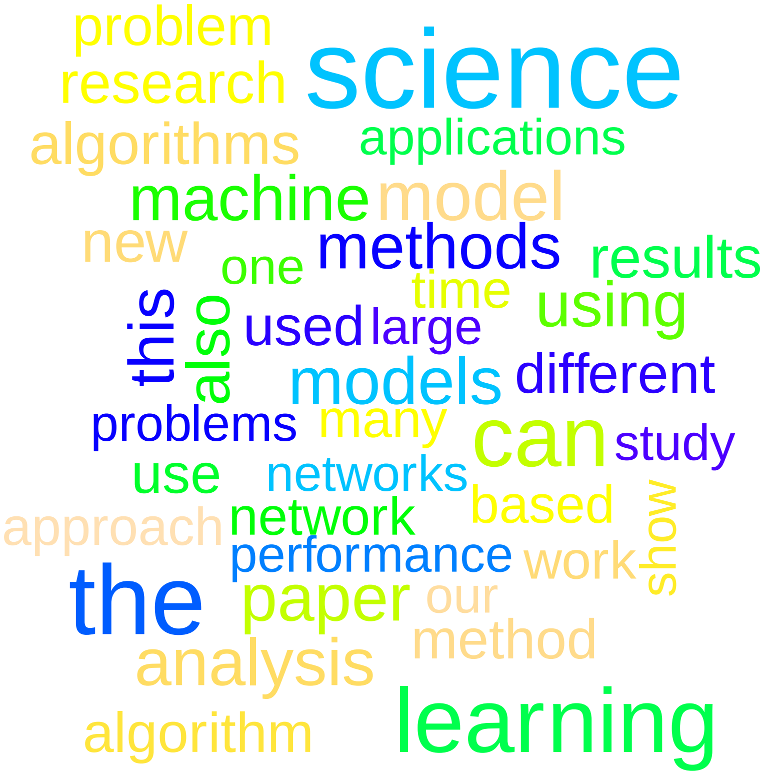
```
## Rows: 1,147
## Columns: 15
## $ id               <chr> "astro-ph/0701361v1", "0901.2805v1", "0901.3118v2"...
## $ submitted        <chr> "2007-01-12 03:28:11", "2009-01-19 10:38:33", "200...
## $ updated          <chr> "2007-01-12 03:28:11", "2009-01-19 10:38:33", "200...
## $ title            <chr> "How to Make the Dream Come True: The Astronomers'...
## $ abstract         <chr> "  Astronomy is one of the most data-intensive of ...
## $ authors          <chr> "Ray P Norris", "Heinz Andernach", "O. V. Verkhoda...
## $ affiliations     <chr> "", "", "Special Astrophysical Observatory, Nizhni...
## $ link_abstract    <chr> "http://arxiv.org/abs/astro-ph/0701361v1", "http:/...
## $ link_pdf         <chr> "http://arxiv.org/pdf/astro-ph/0701361v1", "http:/...
## $ link_doi         <chr> "", "http://dx.doi.org/10.2481/dsj.8.41", "http://...
## $ comment          <chr> "Submitted to Data Science Journal Presented at CO...
## $ journal_ref      <chr> "", "", "", "", "EPJ Data Science, 1:9, 2012", "",...
## $ doi              <chr> "", "10.2481/dsj.8.41", "10.2481/dsj.8.34", "", "1...
## $ primary_category <chr> "astro-ph", "astro-ph.IM", "astro-ph.IM", "astro-p...
## $ categories       <chr> "astro-ph", "astro-ph.IM|astro-ph.CO", "astro-ph.I...
```

```r
DataSciencePapers <- DataSciencePapers %>%
  mutate(
    submitted = lubridate::ymd_hms(submitted)
    , updated = lubridate::ymd_hms(updated)
    , field = str_extract(primary_category, "^[a-z,-]+")
    , Field = ifelse(field == "cs", "Computer Science", "Other discipline"))

# stop words dataset provided by the tidytext package
stop_words <- tidytext::stop_words

view(stop_words)

# the unnest_tokens function is from the tidytext package
arxiv_words <- DataSciencePapers %>%
  unnest_tokens(output = word, input = abstract, token = "words") %>%
  anti_join(stop_words, by = "word") %>%
  select(word, id)

arxiv_word_freqs <- arxiv_words %>%
  count(id, word, sort = TRUE) %>%
  select(word, n, id)


# the wordcloud function is from the wordcloud package
# you may also need to install the "tm" package in order to use the function
set.seed(1962)
wordcloud(DataSciencePapers$abstract,
  max.words = 40,
  scale = c(8, 1),
  colors = topo.colors(n = 30),
  random.color = TRUE)
```

```
## Loading required namespace: tm

## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
## Warning in tm_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,
## tm::stopwords())): transformation drops documents
```

```
## Warning in wordcloud(DataSciencePapers$abstract, max.words = 40, scale = c(8, :
## information could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(DataSciencePapers$abstract, max.words = 40, scale = c(8, :
## data could not be fit on page. It will not be plotted.
```


problem
science
research
algorithms
applications
machine
model
new
methods
results
one
time
this
also
used
large
using
models
different
problems
many
can
study
use
networks
based
approach
network
show
performance
work
the
paper
our
method
analysis
learning
algorithm