STAT 231: Problem Set 5A

Mike Santos

due by 2 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: The first call returns each line number which corresponds with Macbeth's lines. In the penultimate incident, the 6 returned records are each line spoken by Macbeth. And the final call, returns the indices of what matches the call.

```
# 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"
```

```
length(macbeth)
```

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

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

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

```
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."
```

```
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: Including the backslashes changes the identity of "." from a metacharacter to a period.

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 a character set, it will search for MAC... where the character following MAC is any capital letter between B and Z.

```
# (1) `str_detect(string=, pattern=)` is equivalent to `grepl(pattern=, x=)`
 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
```

```
# (2) `str_which(string=, pattern=)` is equivalent to `grep(pattern=, x=)`
 head(grep("MAC[B-Z]", macbeth))
```

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

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

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

```r
# (3) `str_subset(string=, pattern=)` is equivalent to `grep(pattern=, x=, value=TRUE)`
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"
```

```r
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"
```