Homework 5: PDF Extracting with Regular Expressions

Assignment

In this homework assignment, you will be scraping text from a crime-log. The goal of the homework assignment is to completely scrape the PDF into a cleaned tibble ready for analysis. While we obviously cannot do analysis with such a small amount of data, you could imagine scaling this up by looping through multiple PDFs. You will need to be somewhat familiar with regular expressions to do this homework assignment. If you need a refresher (as many of us do), the R for Data Science Textbook and the Guided Exercises are a great reference. Extracting PDFs requires a mastery of the `tidyr::extract` function.

Coding Assignment

0. Save the pdf you downloaded from Gauchospace as `1-20-16.pdf`.

1. Import the PDF using the `pdftools::pdf_text` function. Save the text as crime_log.

2. Using `stringr::str_split`, `base::unlist`, `stringr::str_trim`, and `stringr:str_to_lower` put the PDF text into a vector that has 1 line of the PDF for each element. See this resource (apologies for my shameful self-promoting) if you need a reference. Save this as a vector named crime_log_text.

3. We need to create vectors that have the indexes of the text lines that we want to extract information from. The following sub-questions will all be similar in nature. In later homework, you will see that we could reduce our amount of time coding (and potential mistakes) by creating a general function for this task.

    a) Create a vector named date_reported_indices using `stringr::str_detect` and `base::which` that is a vector of the indices that begin with the words "date reported".
    b) Create a vector named location_indices using `stringr::str_detect` and `base::which` that is a vector of the indices that begin with the words "general location".
    c) Create a vector named date_occurred_from_indices using `stringr::str_detect` and `base::which` that is a vector of the indices that begin with the words "date occurred from".
    d) Create a vector named using `stringr::str_detect` and `base::which` that is a vector of the indices that begin with the words "date occurred to".
    e) Create a vector named incident_indices using `stringr::str_detect` and `base::which` that is a vector of the indices that begin with the words "incident".
    f) Create a vector named disposition_indices using `stringr::str_detect` and `base::which` that is a vector of the indices that begin with the word "disposition".
    g) Create a vector named modified_indices using `stringr::str_detect` and `base::which` that is a vector of the indices that begin with the word "modified".

4. Now we need to extract our desired information into tibbles using the `tibble::as_tibble` and `tidyr::extract` function. As with Question 3, you will be doing the same sort of process each time, but really just changing the regular expression in the `tidyr::extract` function. Remember, if you need help, check out this tutorial (self-promoting shame once more). The following are ordered by what should be easiest to hardest regular expressions.

a) Extract a tibble using the disposition_indices vector in conjuction with crime_log_text. Assign the tibble the name disposition. Your tibble should look like Table 1.

Table 1: Final tibble for 4a.

disposition
closed case- arrest
closed case- records only
NA
open case
closed case- no arrest

b) Extract a tibble using the location_indices vector in conjuction with crime_log_text. Assign the tibble the name location. Your tibble should look like Table 2.

Table 2: Final tibble for 4b

location
all other non-university - non-reportable location
assembly hall - on campus
willkie north - on campus - in any student residential facility
alpha phi - non-campus building or property
ashton moffat hall - on campus - in any student residential facility

c) Extract a tibble using the incident_indices vector in conjuction with crime_log_text. Assign the tibble the name incident. Your tibble should look like Table 3.

Table 3: Final tibble for 4c.

incident
driving under the influence
fire alarms - actual, not arson
harassment/intimidation // possession - marijuana
harassment/intimidation
possession - marijuana

d) Extract a tibble using the modified_indices vector in conjuction with crime_log_text. Assign the tibble the name modified. Your tibble should look like Table **??**.

e) Extract a tibble using the date_occurred_from_indices vector in conjuction with crime_log_text. Assign the tibble the name date_occurred_from. Your tibble should look like Table 5.

f) Extract a tibble using the date_occurred_to_indices vector in conjuction with crime_log_text. Assign the tibble the name date_occurred_to. Your tibble should look like Table 6.

Table 4: Final tibble for 4d.

modified_date	modified_time
01/20/16	17:18
01/20/16	15:38
01/26/16	17:21
01/26/16	17:00
01/21/16	09:36

Table 5: Final tibble for 4e.

date_occurred_from	time_occurred_from
01/20/16	01:48
01/20/16	11:55
01/19/16	13:00
01/20/16	11:21
01/20/16	20:27

Table 6: Final tibble for 4f.

date_occurred_to	time_occurred_to
01/20/16	01:49
01/20/16	12:02
01/20/16	13:09
01/20/16	18:47
01/20/16	20:50

Table 7: Final tibble for 4g.

date_reported	time_reported	report_number
01/20/16	01:49	160151
01/20/16	11:57	160152
01/20/16	13:09	160155
01/20/16	18:47	160157
01/20/16	20:26	160158

g) Extract a tibble using the date_reported_indices vector in conjuction with crime_log_text. Assign the tibble the name date_reported. Your tibble should look like Table 7.

5. Using `dplyr::bind_cols`, bind together each of the tibbles created in Question 4. Save this new tibble as final_crime_log.

6. Using the `lubridate` function, change the `date_reported`, `date_occurred_from`, and `date_occurred_to` columns to be of standard format: YYYY-MM-DD. Attempt to do this using `dplyr::across`, as mastering this function can save a lot of time. Save the updated tibble as final_crime_log_cleaned.