NCollin_HW9_607

Noah Collin

10/23/2021

Homework 9

Noah Collin DATA 607

Libraries:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(httr)
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'

## The following object is masked from 'package:purrr':
##
##     flatten
```

API Key:

For this homework, I'll hard code in my API Key. If this were to be more "reproducible", I might require a user to enter their API key.

```
#https://api.nytimes.com/svc/search/v2/articlesearch.json?q=election&api-key=yourkey

myapiKey <-  "dddddddddddddddddddddddddddddddd"

print("You're using my API key here.  Please consider using your own.  If you have one, enter it here.
## [1] "You’re using my API key here.  Please consider using your own.  If you have one, enter it here.
```

```
UserInput <- ""
UserInput <- readline();
```

```
library(stringr)
if (str_length(myapiKey)  == str_length(UserInput)) {
  myapiKey <- UserInput
  print("Thanks for using your own API key.")
} else {
  print("You're using my key still.  Please don't overuse it...")
}
```

```
## [1] "You’re using my key still.  Please don’t overuse it..."
```

API Variables:

I’m going to search for the term *stenographer*. If you’re so inclined, you can change the search term, sort by newest, or change the year of search with the variables below. Just un-comment out the comments and make them strings. The filters tend to yield less interesting results honestly.

```
searchTerm <- "stenographer"
sortNewest <- "" #"&sort=newest"
filter1 <- "" #"&fq=pub_year:(2008)"
```

Calling the API:

```
NYTAPIstring1 <- str_c( "https://api.nytimes.com/svc/search/v2/articlesearch.json?q=",
                        searchTerm,
                        filter1,
                        sortNewest,
                        "&api-key=",
                        myapiKey)

NYTCall1 <- GET(NYTAPIstring1,
                verbose())

details <- content(NYTCall1,
                   "parse")
```

Making the DataFrame:

```
headlines <- c()
snippets <- c()
wordCounts <- c()
URLs <- c()
for (i in seq(1:10)) {
  if (length(headlines) > i-1) {
    headlines <- c()
  }
   if (length(snippets) > i-1){
    snippets <- c()
   }
   if (length(wordCounts) > i-1) {
    wordCounts <- c()
   }
     if (length(URLs) > i-1) {
    URLs <- c()
  }

  headlines <- c(headlines,details$response$docs[[i]]$headline$main)
  snippets <- c(snippets,details$response$docs[[i]]$snippet)
  wordCounts <- c(wordCounts,details$response$docs[[i]]$word_count)
  URLs <- c(URLs, details$response$docs[[i]]$web_url)
}

#print(length(snippets))

df1 <- data_frame("headlines" = headlines, "snippets" = snippets, "Reported Word Count" = wordCounts, "
```

```
## Warning: ‘data_frame()‘ was deprecated in tibble 1.1.0.
## Please use ‘tibble()‘ instead.
```

Printing the DataFrame:

The dataframe has four relevant columns populated with the correct information. Please note that the WordCount column seems to give odd results from the NYT API.

```
print(df1)
```

```
## # A tibble: 0 x 0
```