Week 9 - Assignment

Vishal Arora, Samriti Malhotra

March 31, 2019

Contents

Introduction **1**

Problem Statment **1**
Solution 1

Summary **4**

Introduction

As part of week9 assignment , we need to connec to one of the New York Times APIs, construct an interface in R to read in the JSON data, and transform it to an R dataframe. -NYT API's home page

Problem Statment

Connect to NYT Article search API, and search for keywords "brexit+deal" , and do some analysis.

Solution

Before connecting to NYT Search API, we need to register ourselves and create an account and then need to generate and Key-API which will be passed along with every request for our search of the article on NYT site. -How to create an account and generate API-KEY on NYT API.

Libraries used in our solution.
- RJSONIO
- RCURL
- dplyr
- kableExtra
- DT

Step1 :- connect to NYT article search and use keywords "brexit+deal" to look for only page 1 as search results and also passing the API-KEY to connect to NYT Api's.
Step2 :- Using getURL function to download the response and then using the fromJSON function from RJSONIO package , we read content in json format and then deserializes to R object.
Step3 :- Using append function to append the output from unlisting till value of docs element in JSON format and append to dat object which is an empty list.
Step4 :- Using as.data.frame object to convert list object dat to Data Frame and using tail to print the last 6 values and dim to print how many observations and variables are present in our data frame object.

```
api <- "bIdanG9zYkeBTahBbLFfiHpZiBbIqmLz"
q <- "brexit+deal"
dat <- c()
uri=paste0("http://api.nytimes.com/svc/search/v2/articlesearch.json?q=brexit+deal&page=1&api-key=bIdanG
d <- getURL(uri)
res <- fromJSON(d,simplify = FALSE)
dat <- append(dat,unlist(res$response$docs))
```

```
df <- as.data.frame(table(dat))
kable(tail(df)) %>%
  kable_styling(bootstrap_options = c("striped","condensed","responsive"),full_width   = F,position = "
  row_spec(0, background ="gray")
```

	dat
86	The Latest: UK Speaker: Govt Can't Present Same Brexit Deal
87	The speaker of Britain's House of Commons dealt a potentially fatal blow to Prime Minister Theres
88	UK's May Urges Lawmakers to Back Her Brexit Deal Now
89	UK PM's May's Brexit Deal Is 'Rancid': Conservative Lawmaker Francois
90	UK Speaker Stymies PM May's Bid for 3rd Vote on Brexit Deal
91	World

```
dim(df)
```

```
## [1] 91  2
```

Step1:- Using the same keyword to search from NYT Article Search API, and instead of only 1 page , setting the size to 500 and using for loop to iterate over each page and gathering the response data based on limit by using fl.Then converting the dates to a vector and append to dat1 object using append function .

> fl = You can limit the number fields returned in the response with the fl parameter. - NYT API Docs for parameters

Step2:- Using strtime function to convert date in charcter format to PSIX format for calendar dat andtime.Then calculation daterange using the min date - max date.
Step3:- Using the seq function to arrange sequentally the dates by day between daterange min date and max date.
Step4:- Aggregate the count for dates into data frame and then compare dates from counts and if on a date there is count then assing 0, otherwise take a count.
Step5:- Before comparing convert POSIX object to character using as.charcter function on strptime.
Step6:- Plot diagram depicting on x-axis date and y-axis with no of article on each date.

```
records <- 300
pageRange <- 0:(records/10-1)
# get data
dat1 <- c()
for (i in pageRange) {
   # concatenate URL for each page
   uri <- paste0("http://api.nytimes.com/svc/search/v2/articlesearch.json?q=", q, "&page=", i, "&fl=pub
   d <- getURL(uri)
   res <- fromJSON(d,simplify = FALSE)
   res
   dat1 <- append(dat1, unlist(res$response$docs))

 }
dat1.conv <- strptime(dat1, format="%Y-%m-%d") # need to convert dat into POSIX format
daterange <- c(min(dat1.conv), max(dat1.conv))
dat1.all <- seq(daterange[1], daterange[2], by="day") # all possible days
cts <- as.data.frame(table(dat1))
dat1.all <- strptime(dat1.all, format="%Y-%m-%d")
kable(head(dat1.all,n=10)) %>%
  kable_styling(bootstrap_options = c("striped","hover","condensed","responsive"),full_width   = F,posi
  row_spec(0, background ="gray")
```

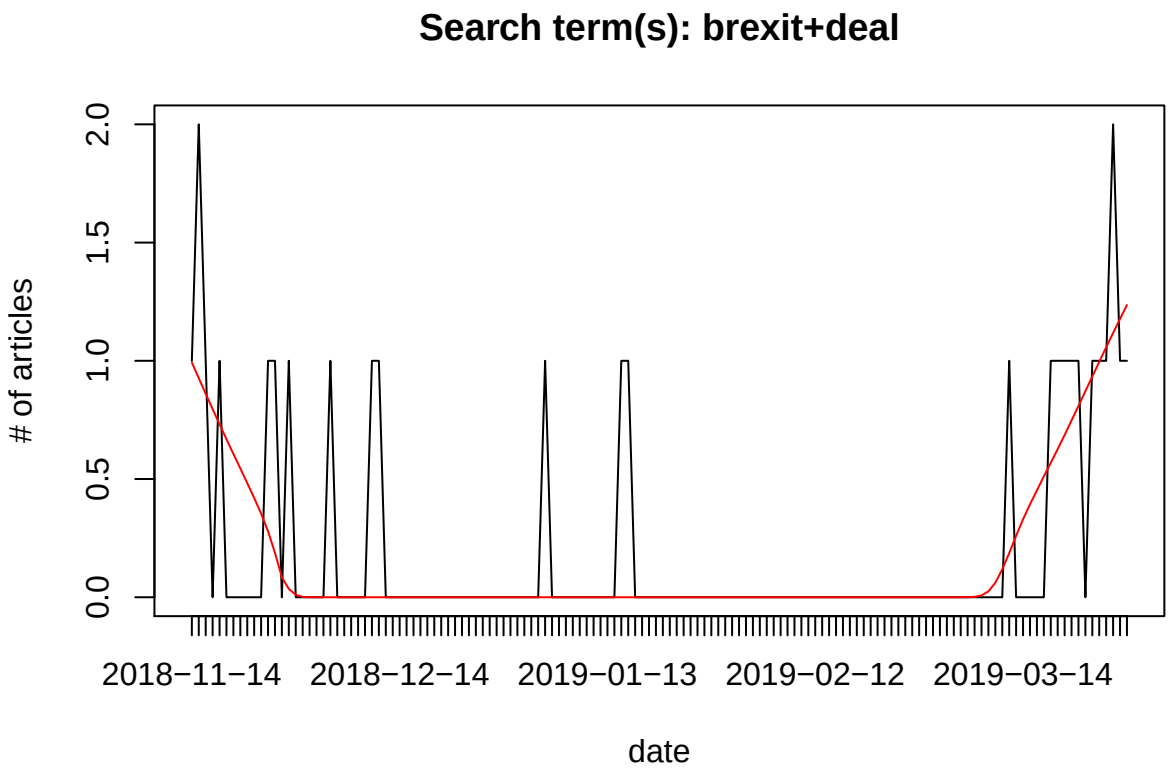
x
2018-11-14
2018-11-15
2018-11-16
2018-11-17
2018-11-18
2018-11-19
2018-11-20
2018-11-21
2018-11-22
2018-11-23

```
freqs <- ifelse(as.character(dat1.all) %in% as.character(strptime(cts$dat1, format="%Y-%m-%d")), cts$Fr

kable(head(freqs)) %>%
  kable_styling(bootstrap_options = c("striped","hover","condensed","responsive"),full_width   = F,posi
  row_spec(0, background ="gray")
```

x
1
2
1
0
1
0

```
plot.default(freqs, type="l", xaxt="n", main=paste("Search term(s):",q), ylab="# of articles", xlab="da
axis(1, 1:length(freqs), dat1.all)
lines(lowess(freqs, f=.2), col = 2)
```



Summary

Looking at the graph, it is clearly visible on what date what is the frequency of articles published in NYT.