LETTERKENNY INSTITUTE OF TECHNOLOGY

ASSIGNMENT COVER SHEET

Lecturer’s Name: James Connolly

Assessment Title: Data scraping

Work to be submitted to: James Connolly

Date for submission of work: 29-08-2018

Place and time for submitting work: 1:00 pm

To be completed by the Student

Student’s Name: PRATEEK PARASHER

Class: Msc in Big Data

Subject/Module: Data Science

Word Count (where applicable):

I confirm that the work submitted has been produced solely through my own efforts.

Student’s signature: PRATEEK PARASHER Date: 28/8/2018

|  |
| --- |
| **Notes**  **Penalties:** The total marks available for an assessment is reduced by 15% for work submitted up to one week late. The total marks available are reduced by 30% for work up to two weeks late. Assessment work received more than two weeks late will receive a mark of zero. [Incidents of alleged plagiarism and cheating are dealt with in accordance with the Institute’s Assessment Regulations.]  **Plagiarism:** Presenting the ideas etc. of someone else without proper acknowledgement (see section L1 paragraph 8).  **Cheating:** The use of unauthorised material in a test, exam etc., unauthorised access to test matter, unauthorised collusion, dishonest behaviour in respect of assessments, and deliberate plagiarism (see section L1 paragraph 8).  **Continuous Assessment:** For students repeating an examination, marks awarded for continuous assessment, shall normally be carried forward from the original examination to the repeat examination. |

**Description**: - Twitter is a great source for sentiment data and social media mining furthermore it is quite easy to get significant amounts of data to be able to scrape data from Twitter. Last year we all experienced the different climate conditions and multiple climate warning announced by govt. this lab I scrapped the data from twitter regarding climate warning like heatwaves warning, snow warning etc. after that I visualize the wordcloud of scraped data from twitter.

GitHub Link :- <https://github.com/prateekparasher/web_scrap>

**STEP DESCRIPTION AND CODE**

**#install package**

install.packages("rvest")

library(rvest)

**#read link**

url <- 'https://twitter.com/weatherrte?lang=en'

web\_page <- read\_html(url)

**#head & str data**

head(web\_page)

str(web\_page)

**#ranking tweet data**

warning <- html\_nodes(web\_page,'.tweet-text')

head(warning, 30)

**#leghth of tweet data**

length(warning)

data <- html\_text(warning)

**#show first 30 tweets**

head(data, 30)

**#Install and load the required packages**

**# Install**

install.packages("tm") # for text mining

install.packages("SnowballC") # for text stemming

install.packages("wordcloud") # word-cloud generator

install.packages("RColorBrewer") # color palettes

**# Load**

library("tm")

library("SnowballC")

library("wordcloud")

library("RColorBrewer")

**# Load the data as a corpus**

docs <- Corpus(VectorSource(data))

#Inspect the content of the document

inspect(docs)

**#Text transformation**

toSpace <- content\_transformer(function (x , pattern ) gsub(pattern, " ", x))

docs <- tm\_map(docs, toSpace, "/")

docs <- tm\_map(docs, toSpace, "@")

docs <- tm\_map(docs, toSpace, "\\|")

**# Convert the text to lower case**

docs <- tm\_map(docs, content\_transformer(tolower))

**# Remove numbers**

docs <- tm\_map(docs, removeNumbers)

**# Remove english common stopwords**

docs <- tm\_map(docs, removeWords, stopwords("english"))

**# Remove your own stop word**

**# stopwords as a character vector**

docs <- tm\_map(docs, removeWords, c("blabla1", "blabla2"))

**# Remove punctuations**

docs <- tm\_map(docs, removePunctuation)

**# Eliminate extra white spaces**

docs <- tm\_map(docs, stripWhitespace)

**# Text stemming**

docs <- tm\_map(docs, stemDocument)

**#Build a term-document matrix**

dtm <- TermDocumentMatrix(docs)

m <- as.matrix(dtm)

v <- sort(rowSums(m),decreasing=TRUE)

d <- data.frame(word = names(v),freq=v)

head(d, 20)

**# Generate the Word cloud**

set.seed(1234)

wordcloud(words = d$word, freq = d$freq, min.freq = 1,

max.words=200, random.order=FALSE, rot.per=0.35,

colors=brewer.pal(8, "Dark2"))

**STR (Structure of scraped data)**

str(dtm)

List of 6

$ i : int [1:355] 1 2 3 4 5 6 7 8 9 10 ...

$ j : int [1:355] 1 1 1 1 1 1 1 1 1 1 ...

$ v : num [1:355] 1 1 1 1 1 1 1 1 1 1 ...

$ nrow : int 222

$ ncol : int 21

$ dimnames:List of 2

..$ Terms: chr [1:222] "can" "changes" "coverage" "forecast" ...

..$ Docs : chr [1:21] "1" "2" "3" "4" ...

- attr(\*, "class")= chr [1:2] "TermDocumentMatrix" "simple\_triplet\_matrix"

- attr(\*, "weighting")= chr [1:2] "term frequency" "tf"

str(d)

'data.frame': 222 obs. of 2 variables:

$ word: Factor w/ 222 levels "across","advisory",..: 153 216 145 208 64 45 52 71 206 154 ...

$ freq: num 12 12 9 9 7 6 6 5 5 4 ...

**RESULT & VISUALIZATION**

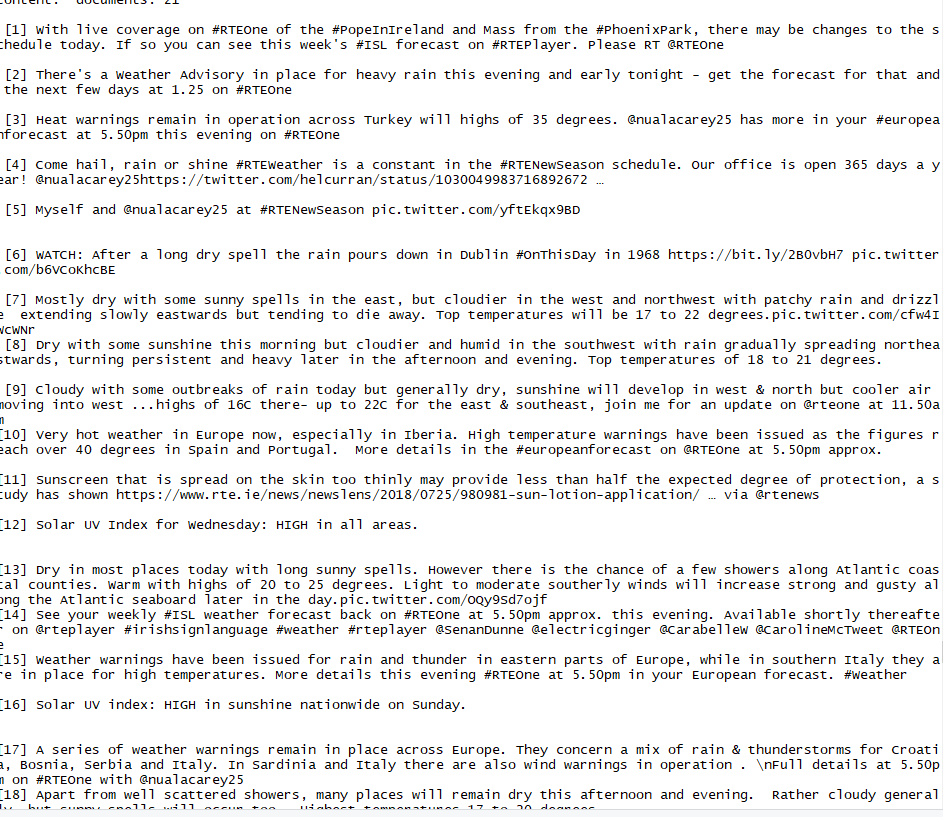


Fig 1 – warning tweets scraped from twitter using R

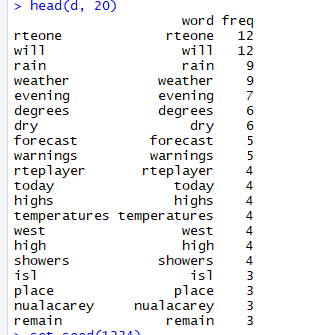


Fig 2 - Table containing the frequency of the words

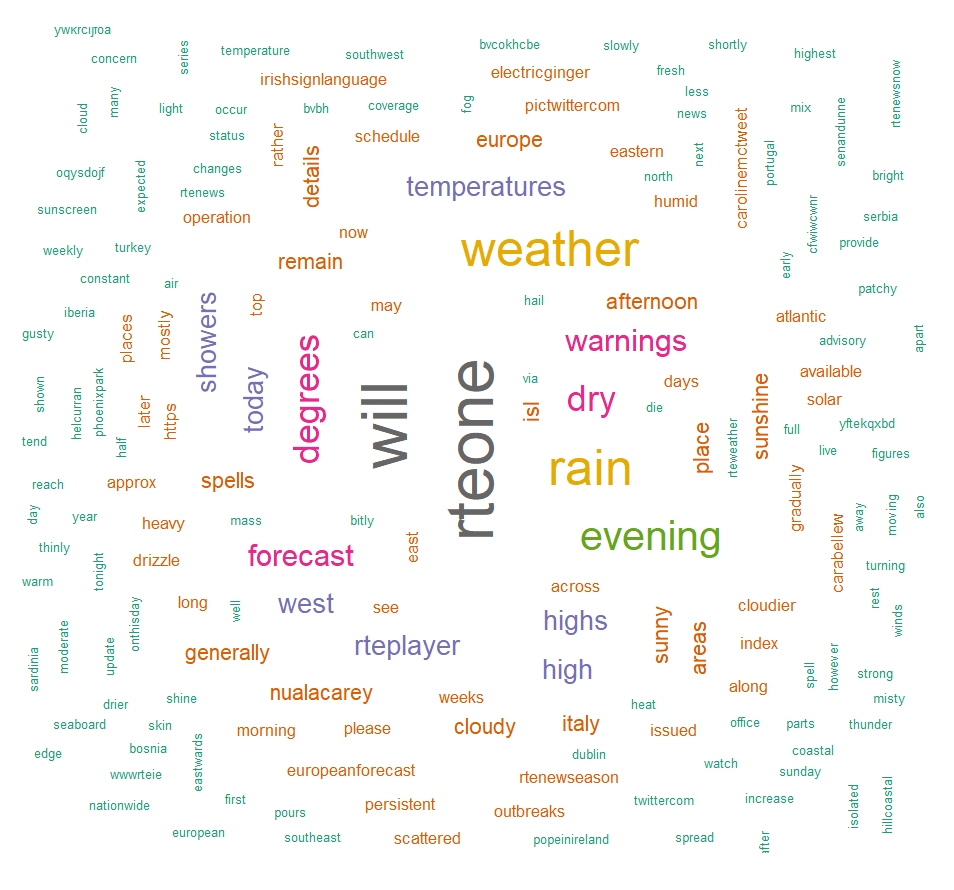


Fig 3 - Data cloud visualization