Name : Gauri N. Baing

RollNo : 509

Class : MSc CS Part II

SNA Innovative Practical

**Mining and Analyzing Tweets using R**

**STEP 1: Getting your Twitter API access**

Apply for a developer account via the following website:  <https://developer.twitter.com/en/apply-for-access.html>.

Once you application has been accepted by Twitter you’ll receive the following credentials that you need to keep safe:

* **Consumer key**
* **Consumer Secret**
* **Access Token**
* **Access Secret**

**STEP 2: Mining Tweets**

Download the package “rtweet”, to extract tweets.

install.packages("rtweet")  
library (rtweet)

Then, set up the authentification to connect to Twitter.

consumer\_key <- "\*\*\*\*"

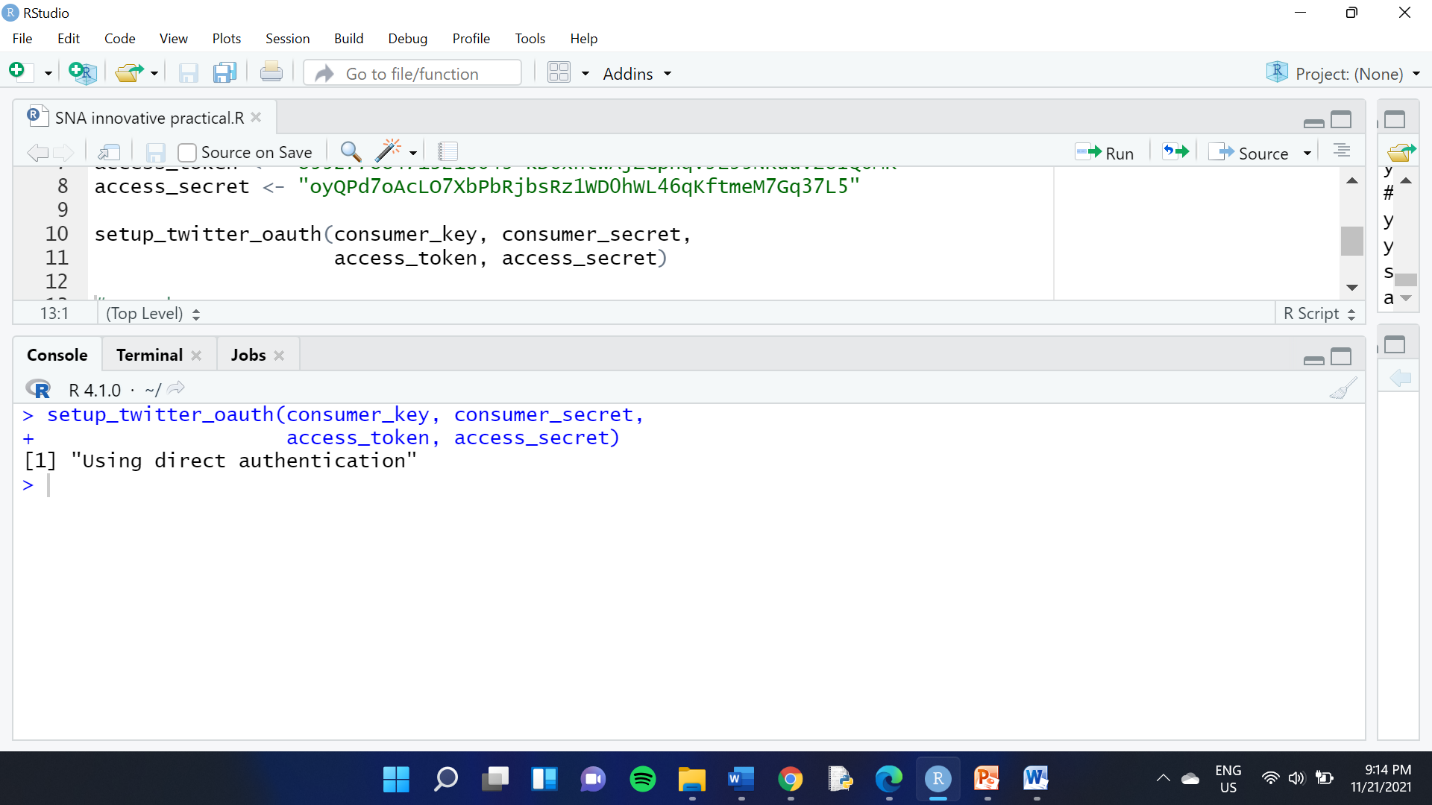
consumer\_secret <- "\*\*\*\*"

access\_token <- "\*\*\*\*"

access\_secret <- "\*\*\*\*"

setup\_twitter\_oauth(consumer\_key, consumer\_secret,

access\_token, access\_secret)

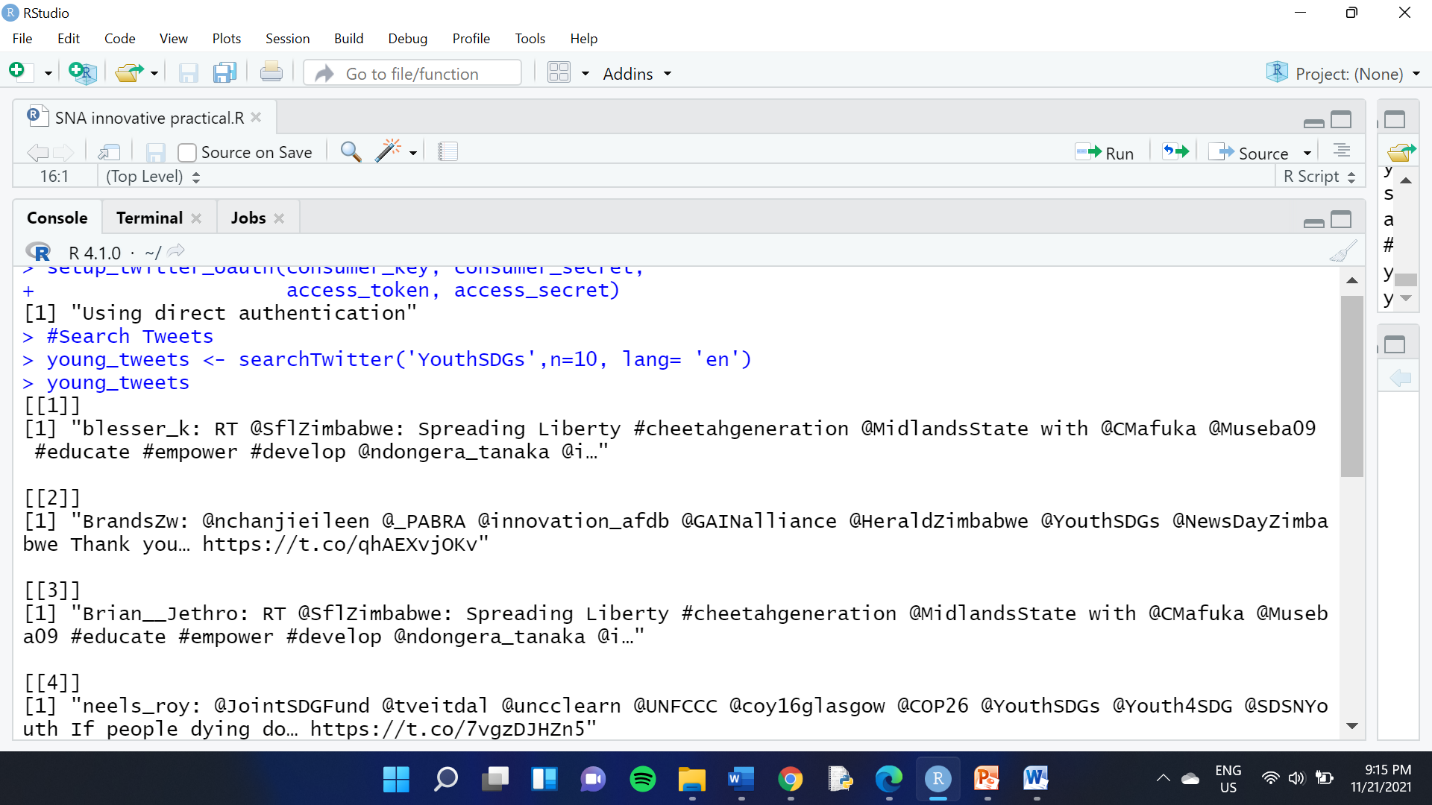


## ****Searching for tweets****

Depending on the analysis you wish to perform, you may want to search for tweets that contain a specific word or hashtag.

Young\_tweets<- search\_tweets(“YouthSDGs”, n=10, lang=”en”)

Young\_tweets



## Search for a specific user account

Gates <- get\_timeline("@BillGates", n= 3200)

# 

# STEP 3: Analyse the tweets

## 1. SHOW WHAT WORKS BEST AND WHAT DOESN’T

The first part of any report should deliver clear information as to what worked best and what didn’t. Finding out the best and least performing tweets gives a quick and clear overall picture of the account.

In order to do this, you first need to distinguish between organic tweets, retweets and replies.

# Remove retweets

Gates\_tweets\_organic <- Gates\_tweets[Gates\_tweets$is\_retweet==FALSE, ]

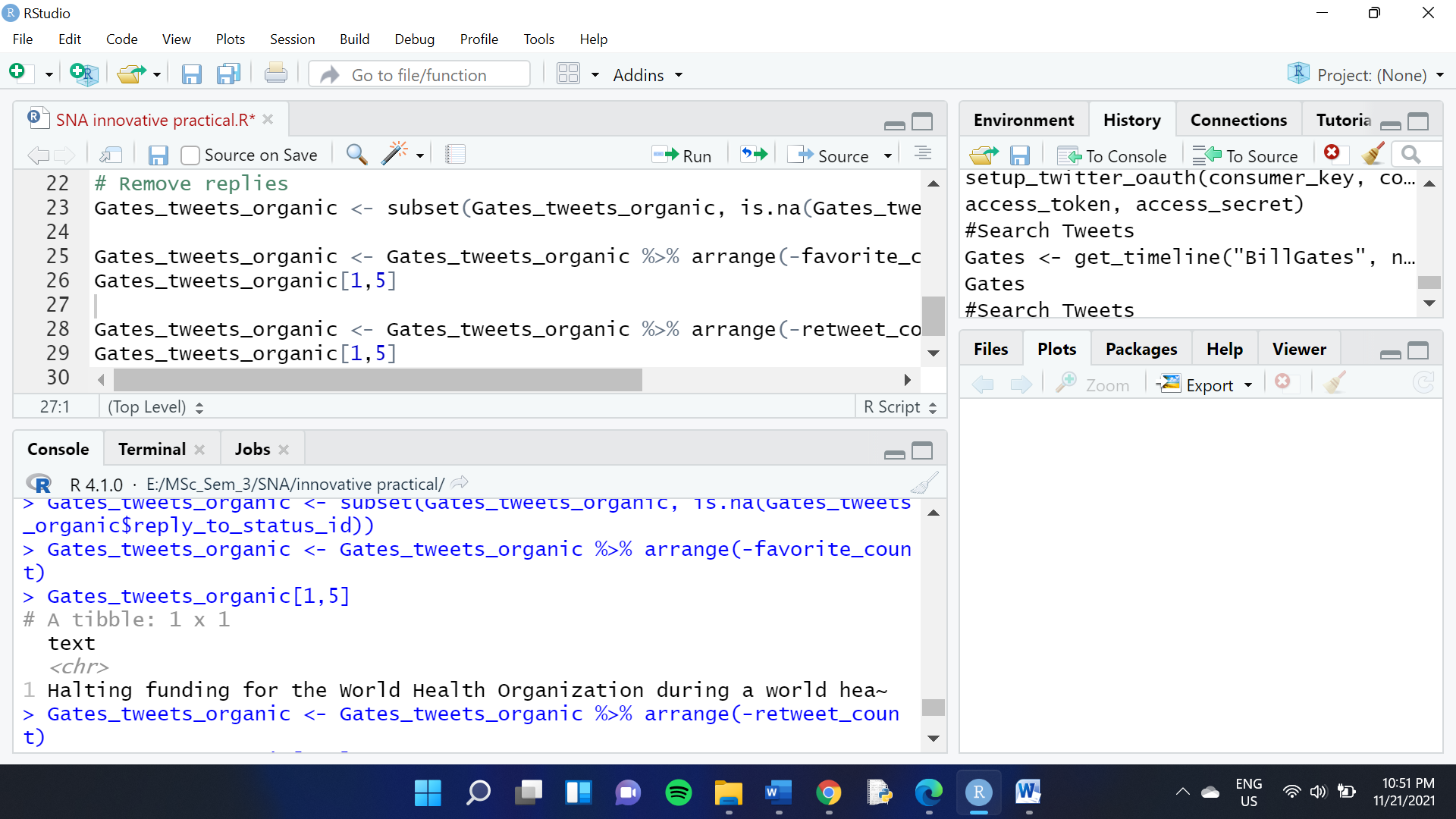
# Remove replies

Gates\_tweets\_organic <- subset(Gates\_tweets\_organic, is.na(Gates\_tweets\_organic$reply\_to\_status\_id))

**favorite\_count** (i.e. the number of likes) or **retweet\_count** (i.e. the number of retweets). Simply arrange them in descending to find the one with the highest number of likes or retweets or ascending order (without the minus) to find the one with lowest number of engagements.

Gates\_tweets\_organic <- Gates\_tweets\_organic %>% arrange(-favorite\_count)  
Gates\_tweets\_organic[1,5]

Gates\_tweets\_organic <- Gates\_tweets\_organic %>% arrange(-retweet\_count)  
Gates\_tweets\_organic[1,5]



## 2. SHOW THE RATIO OF REPLIES/RETWEETS/ORGANIC TWEETS

Analyzing the ratio of replies, retweets and organic tweets can tell you a great deal about the type of account you’re analyzing.

As a first step, make sure to create three different data sets. As we’ve already created a dataset containing only the organic tweets in the previous steps, simply now create a dataset containing only the retweets and one containing only the replies.

# Keeping only the retweets  
Gates\_retweets <- Gates\_tweets[Gates\_tweets$is\_retweet==TRUE,]

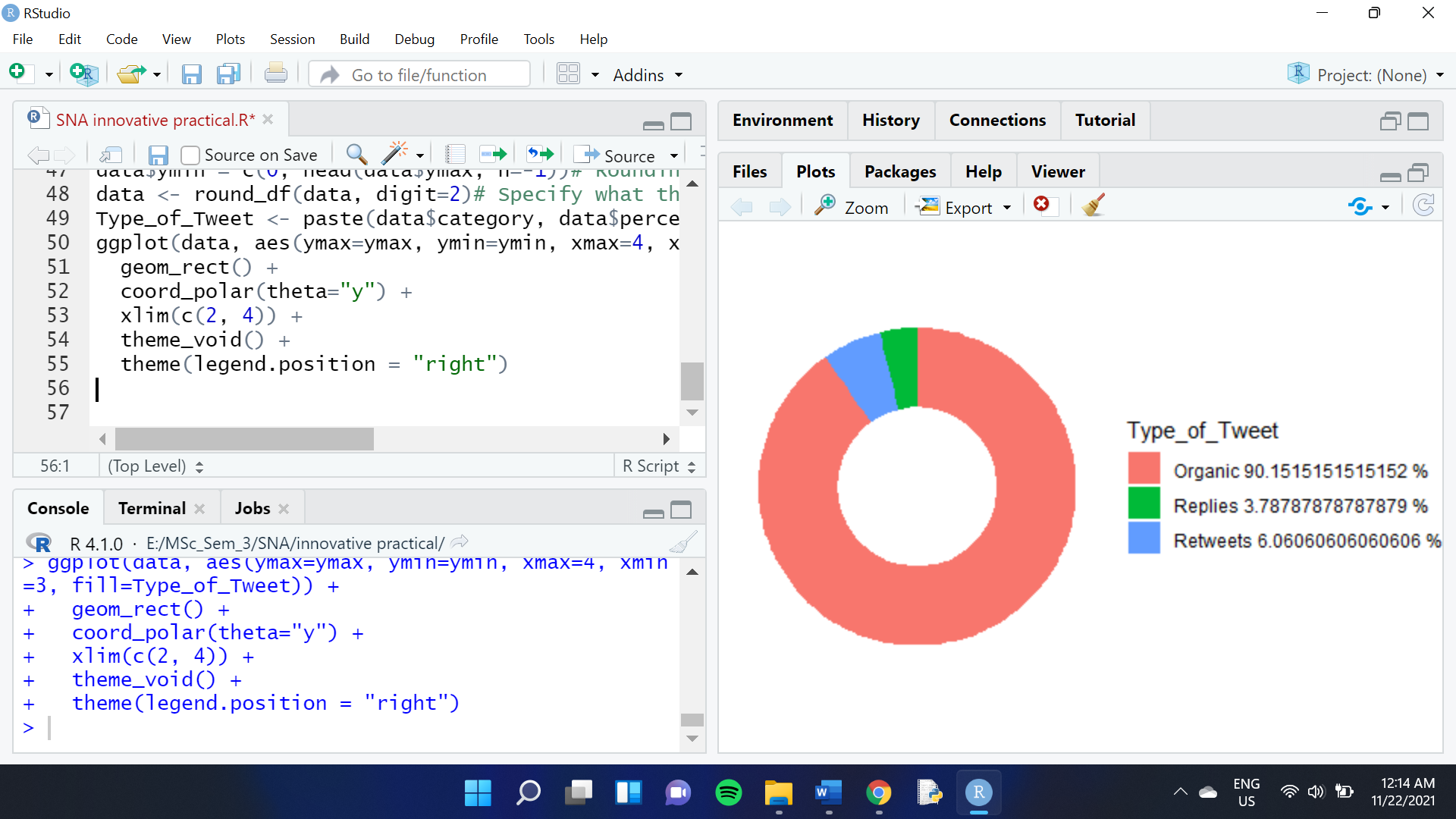
# Keeping only the replies  
Gates\_replies <- subset(Gates\_tweets, !is.na(Gates\_tweets$reply\_to\_status\_id))

Then, create a separate data frame containing the number of organic tweets, retweets, and replies.

# Creating a data frame  
data <- data.frame(  
 category=c("Organic", "Retweets", "Replies"),  
 count=c(2856, 192, 120)  
)

Once done that, we can start preparing your data frame for a donut chart as shown below. This includes adding columns that calculate the ratios and percentages and some visualization tweaks such as specifying the legend and rounding up data.

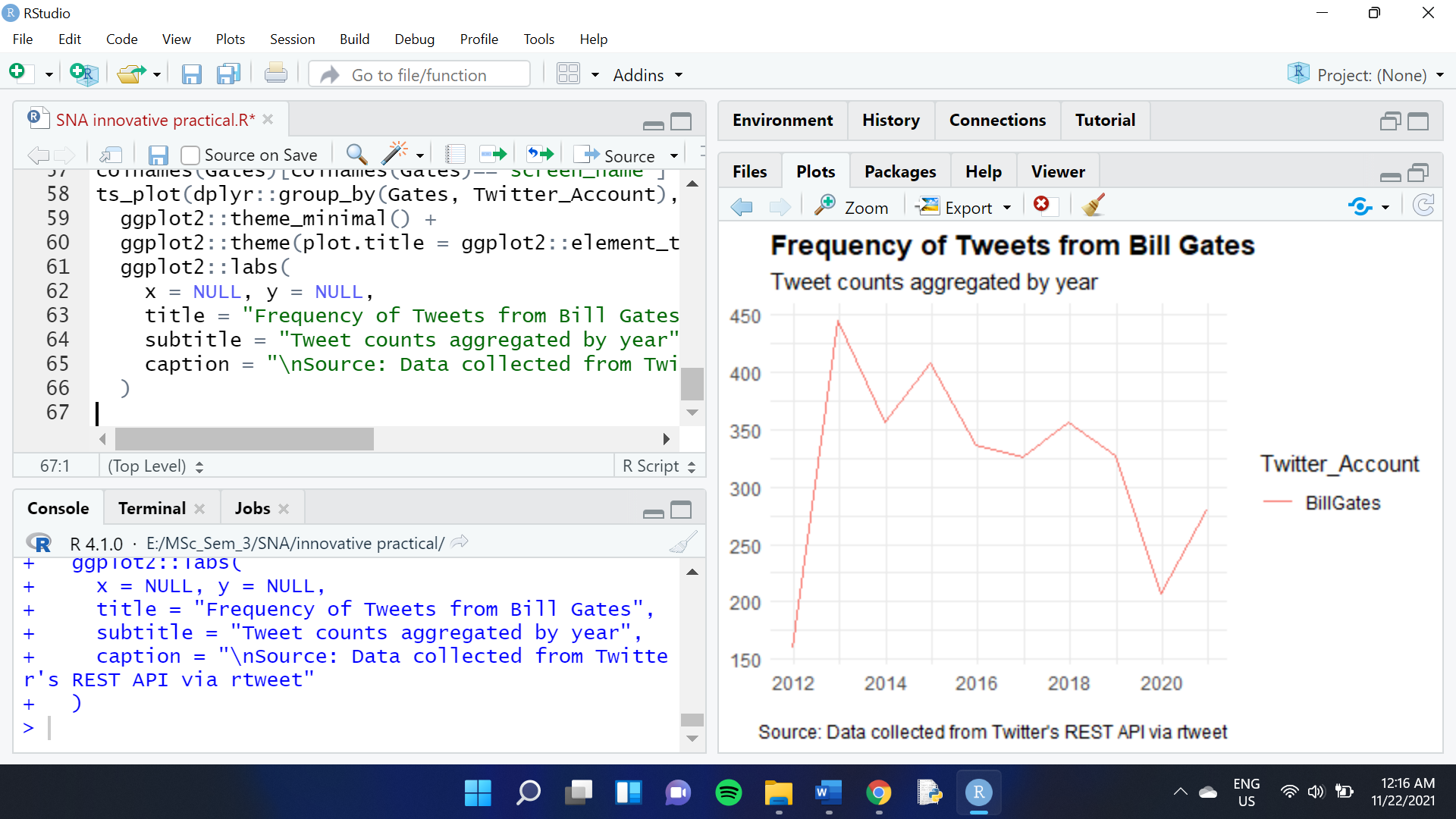
# Adding columns   
data$fraction = data$count / sum(data$count)  
data$percentage = data$count / sum(data$count) \* 100  
data$ymax = cumsum(data$fraction)  
data$ymin = c(0, head(data$ymax, n=-1))# Rounding the data to two decimal points  
data <- round\_df(data, 2)# Specify what the legend should say  
Type\_of\_Tweet <- paste(data$category, data$percentage, "%")ggplot(data, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3, fill=Type\_of\_Tweet)) +  
 geom\_rect() +  
 coord\_polar(theta="y") +   
 xlim(c(2, 4)) +  
 theme\_void() +  
 theme(legend.position = "right")



## 3. SHOW WHEN THE TWEETS ARE PUBLISHED

Thanks to the date and hour extracted with each tweet, understanding when Bill Gates tweets most is very easy to analyze. This can give us an overall overview of the activity of the account and can be a useful metric to be analyzed against the most and least performing tweets.

colnames(Gates\_tweets)[colnames(Gates\_tweets)=="screen\_name"] <- "Twitter\_Account"ts\_plot(dplyr::group\_by(Gates\_tweets, Twitter\_Account), "year") +  
 ggplot2::theme\_minimal() +  
 ggplot2::theme(plot.title = ggplot2::element\_text(face = "bold")) +  
 ggplot2::labs(  
 x = NULL, y = NULL,  
 title = "Frequency of Tweets from Bill Gates",  
 subtitle = "Tweet counts aggregated by year",  
 caption = "\nSource: Data collected from Twitter's REST API via rtweet"  
 )



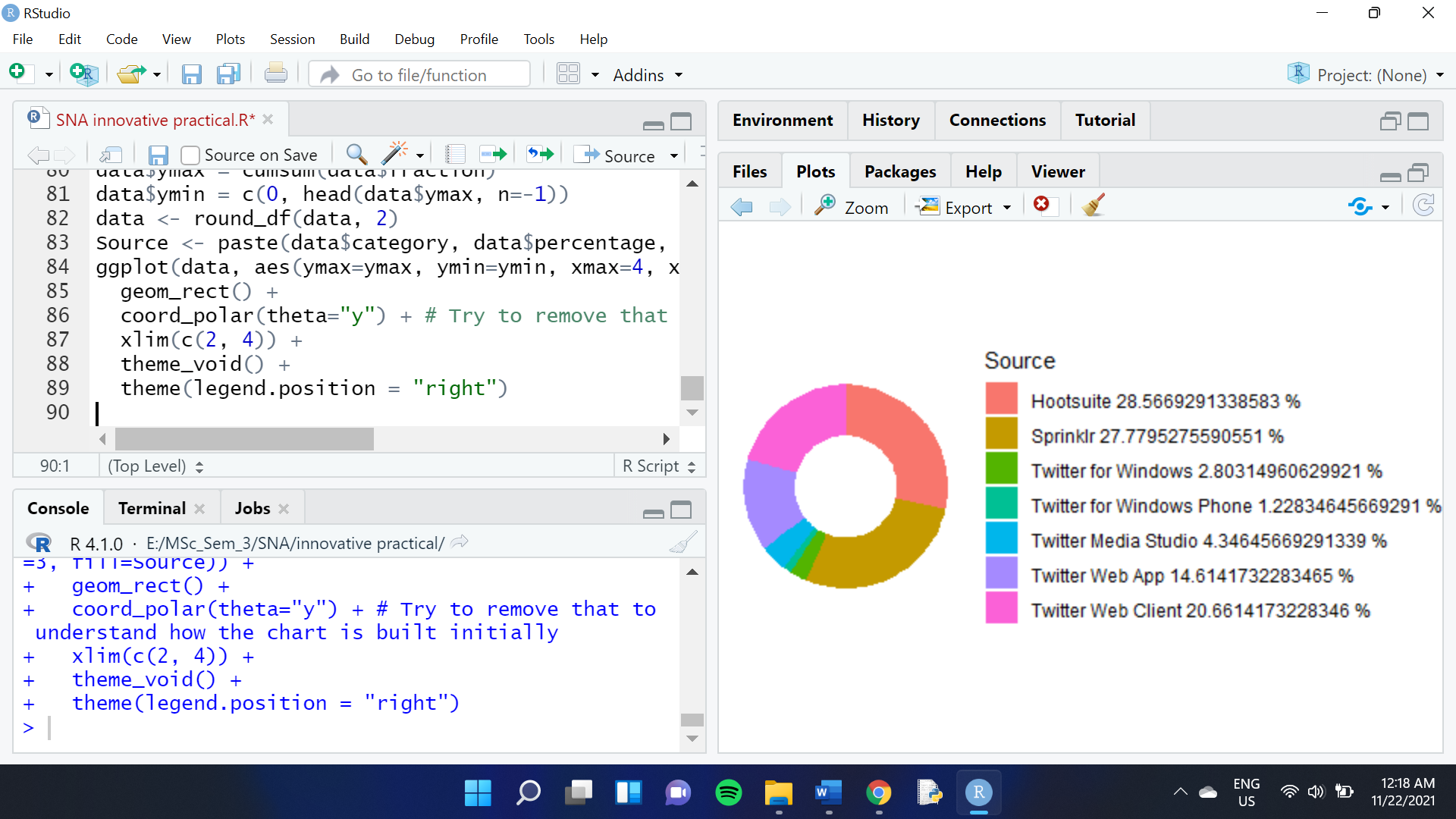
## 4. SHOW FROM WHERE THE TWEETS ARE PUBLISHED

Analyzing the source of the platform from which tweets are published is another cool insight to have. One of the reasons is that we can to a certain extent deduct whether or not Bill Gates is the one tweeting or not. As a result, this helps us define the personality of the tweets.

Gates\_app <- Gates\_tweets %>%   
 select(source) %>%   
 group\_by(source) %>%  
 summarize(count=n())Gates\_app <- subset(Gates\_app, count > 11)

Once this is done, the process is similar to the donut chart .

data <- data.frame(  
 category=Gates\_app$source,  
 count=Gates\_app$count  
)data$fraction = data$count / sum(data$count)  
data$percentage = data$count / sum(data$count) \* 100  
data$ymax = cumsum(data$fraction)  
data$ymin = c(0, head(data$ymax, n=-1))data <- round\_df(data, 2)Source <- paste(data$category, data$percentage, "%")ggplot(data, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3, fill=Source)) +  
 geom\_rect() +  
 coord\_polar(theta="y") + # Try to remove that to understand how the chart is built initially  
 xlim(c(2, 4)) +  
 theme\_void() +  
 theme(legend.position = "right")



Note that most of the tweets from Bill Gates originate from Twitter Web Client, Sprinklr and Hootsuite — an indication that Bill Gates is most likely not the one tweeting himself!

## 5. SHOW THE MOST FREQUENT WORDS FOUND IN THE TWEETS

A Twitter Analytics Report should of course include an analysis of the content of the tweets and this includes finding out which words are used most.

Because you’re analyzing textual data, make sure to clean it first and remove it from any character that you don’t want to show in your analysis such as hyperlinks, @ mentions or punctuations.

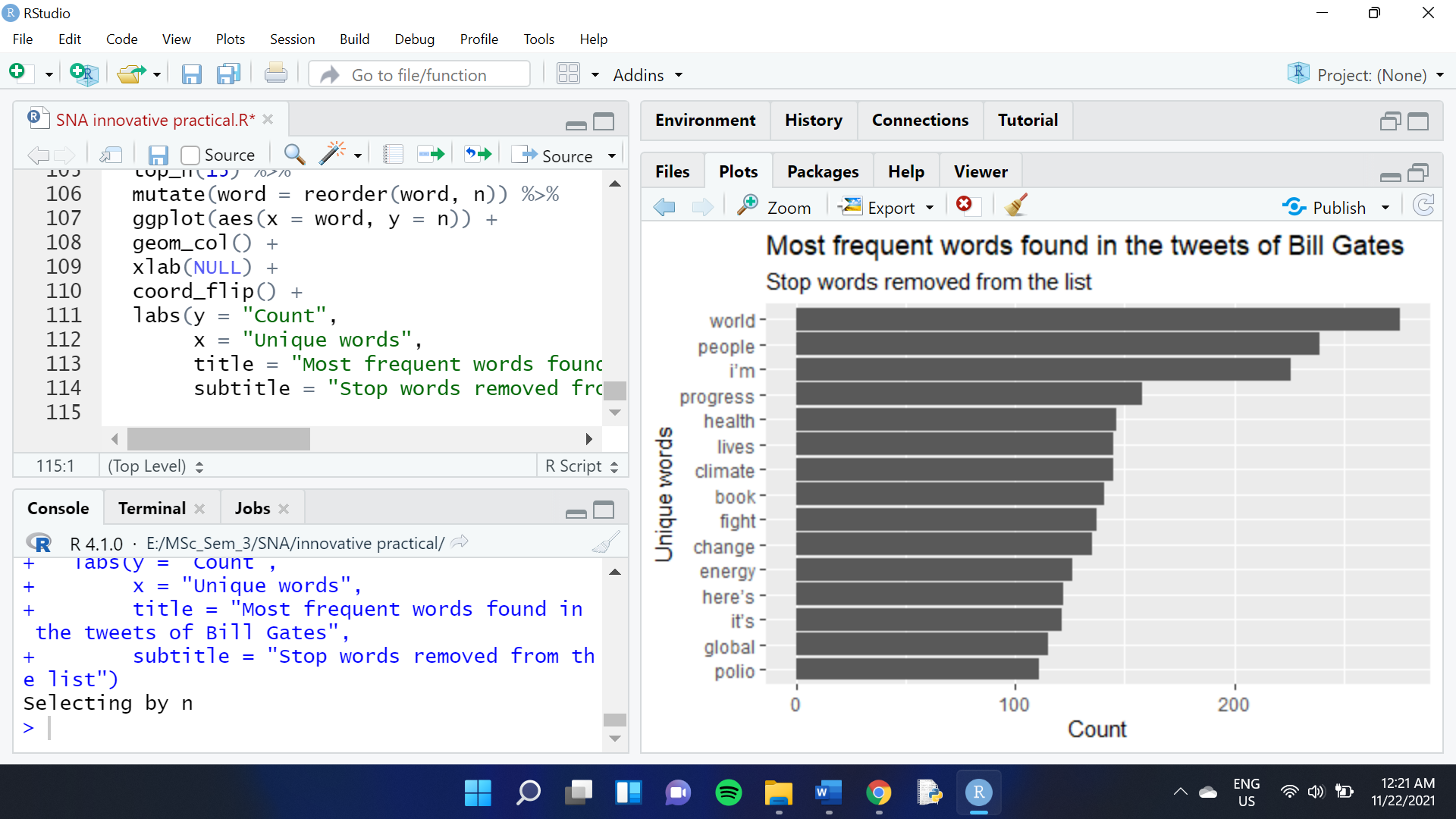
Gates\_tweets\_organic$text <- gsub("https\\S\*", "", Gates\_tweets\_organic$text)Gates\_tweets\_organic$text <- gsub("@\\S\*", "", Gates\_tweets\_organic$text) Gates\_tweets\_organic$text <- gsub("amp", "", Gates\_tweets\_organic$text) Gates\_tweets\_organic$text <- gsub("[\r\n]", "", Gates\_tweets\_organic$text)Gates\_tweets\_organic$text <- gsub("[[:punct:]]", "", Gates\_tweets\_organic$text)

As a second step, make sure to remove stop words from the text. This is important for your analysis of the most frequent words as you don’t want the most common used words such as “to” or “and” to appear as these don’t carry much meaning for your analysis.

tweets <- Gates\_tweets\_organic %>%  
 select(text) %>%  
 unnest\_tokens(word, text)tweets <- tweets %>%  
 anti\_join(stop\_words)

You can then plot the most frequent words found in the tweets by following the simple steps below.

tweets %>% # gives you a bar chart of the most frequent words found in the tweets  
 count(word, sort = TRUE) %>%  
 top\_n(15) %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(x = word, y = n)) +  
 geom\_col() +  
 xlab(NULL) +  
 coord\_flip() +  
 labs(y = "Count",  
 x = "Unique words",  
 title = "Most frequent words found in the tweets of Bill Gates",  
 subtitle = "Stop words removed from the list")



## 6. PERFORM A SENTIMENT ANALYSIS OF THE TWEETS

Finally, you may want to add a sentiment analysis at the end of your Twitter Analytics Report. This is easy to do with the package “syuzhet” and allows you to further deepen your analysis by grasping the tone of the tweets. No one likes a Twitter account that only spreads angry or sad tweets. Capturing the tone of your tweets and how they balance out is a good indication of your account’s performance.

library(syuzhet)# Converting tweets to ASCII to trackle strange characters  
tweets <- iconv(tweets, from="UTF-8", to="ASCII", sub="")# removing retweets, in case needed   
tweets <-gsub("(RT|via)((?:\\b\\w\*@\\w+)+)","",tweets)# removing mentions, in case needed  
tweets <-gsub("@\\w+","",tweets)ew\_sentiment<-get\_nrc\_sentiment((tweets))  
sentimentscores<-data.frame(colSums(ew\_sentiment[,]))names(sentimentscores) <- "Score"sentimentscores <- cbind("sentiment"=rownames(sentimentscores),sentimentscores)rownames(sentimentscores) <- NULLggplot(data=sentimentscores,aes(x=sentiment,y=Score))+  
 geom\_bar(aes(fill=sentiment),stat = "identity")+  
 theme(legend.position="none")+  
 xlab("Sentiments")+ylab("Scores")+  
 ggtitle("Total sentiment based on scores")+  
 theme\_minimal()

