Sentimental Analysis, Most Frequent Hashtags Used In A Country and Wordcloud Formation using R

FROM TWITTER DATA

******

# Introduction

***This project aims to do a Sentiment Analysis of Twitter data on BJP.***

*Elections of 2019 are around the corner. Everyone is eager to know who will lead the country for the next 5 years. Here is a Sentimental analysis on the political party BJP which will give us an idea of how people think about the party i.e. if people have a positive take on it or a negative take.*

**R libraries needed to achieve this task :**

* ROAuth
* twitteR
* ggplot2
* sentimentr
* wordcloud
* syuzhet
* dplyr
* plyr
* stringr

# Overview

Tweets are imported using R and the data is cleaned by removing emoticons and URLs. Lexical Analysis is used to predict the sentiment of tweets and subsequently express the opinion graphically through ggplots, histogram and tables.

# Steps performed in the project

1. Extraction of Tweets
2. Storing Tweets in an Excel sheet
3. Cleaning Tweets
4. Loading Word Database and Lexical Analysis
5. Finding Trends of a country
6. Wordcloud Formation

# Step 1 : Extraction of Tweets

(i) Create twitter application

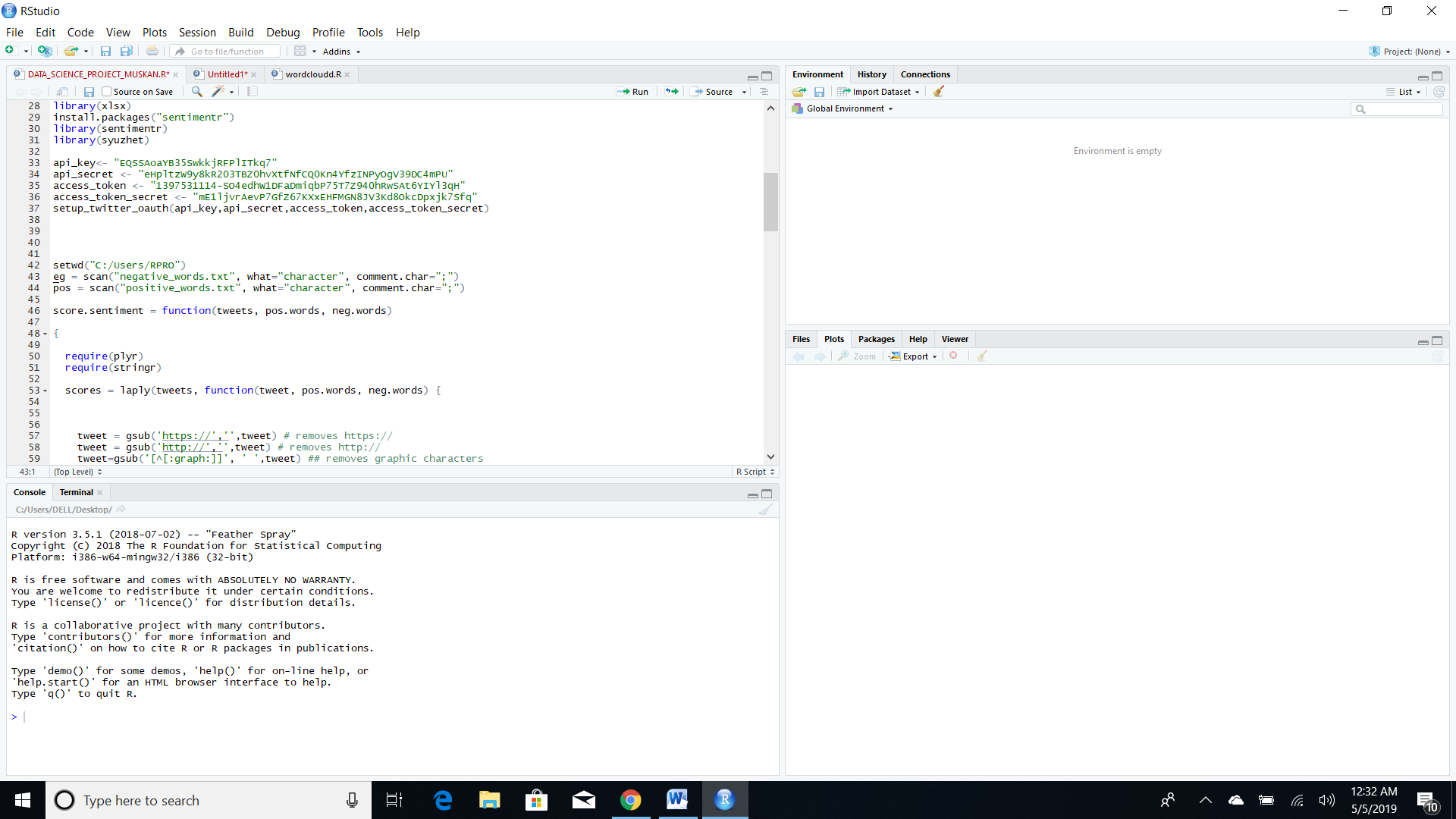
(ii) twitteR - Provides an interface to the Twitter web API

(iii) ROAuth- ​R Interface For OAuth

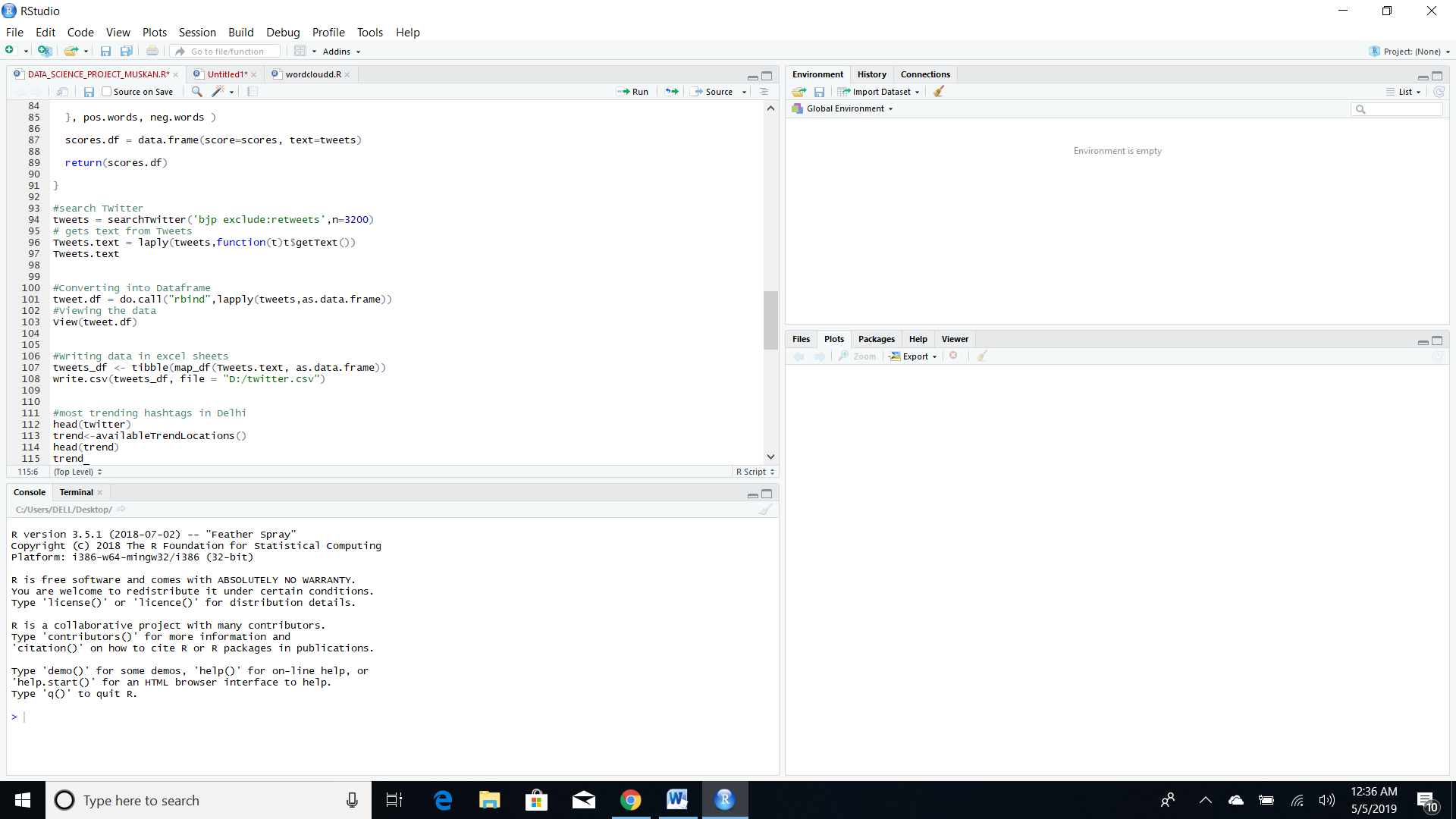
(iv)​Create twitter authenticated credential object, It is done using consumer key, consumer secret, access token, access secret.

(v) During authentication, we are redirected to a URL automatically where we click on Authorize app as shown in the image below and enter the unique 7-digit number to get linked to the account from which feeds are being taken.

​



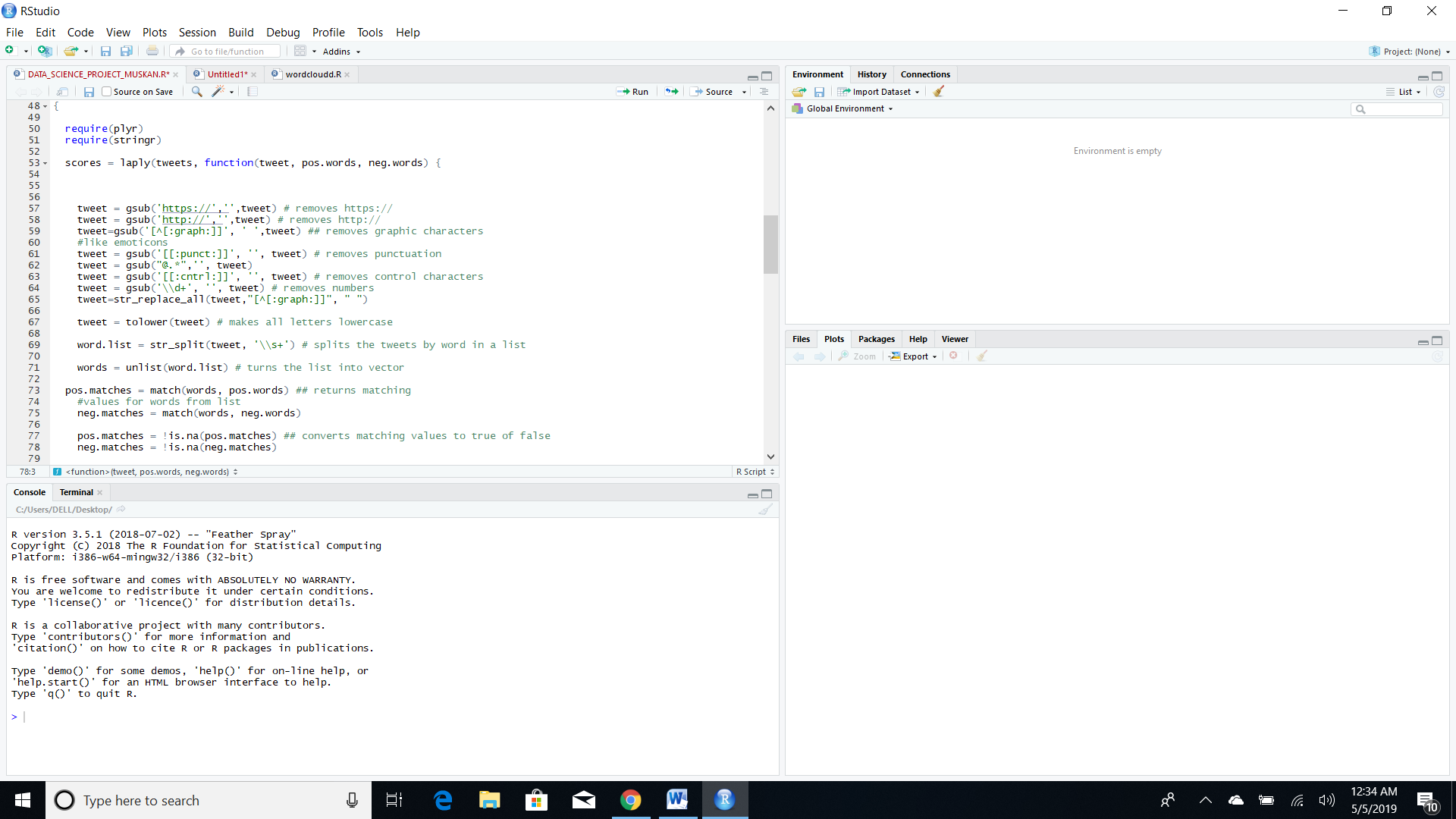
# Step 2 :Storing Tweets in an Excel sheet



# Step 3 : Cleaning of Tweets

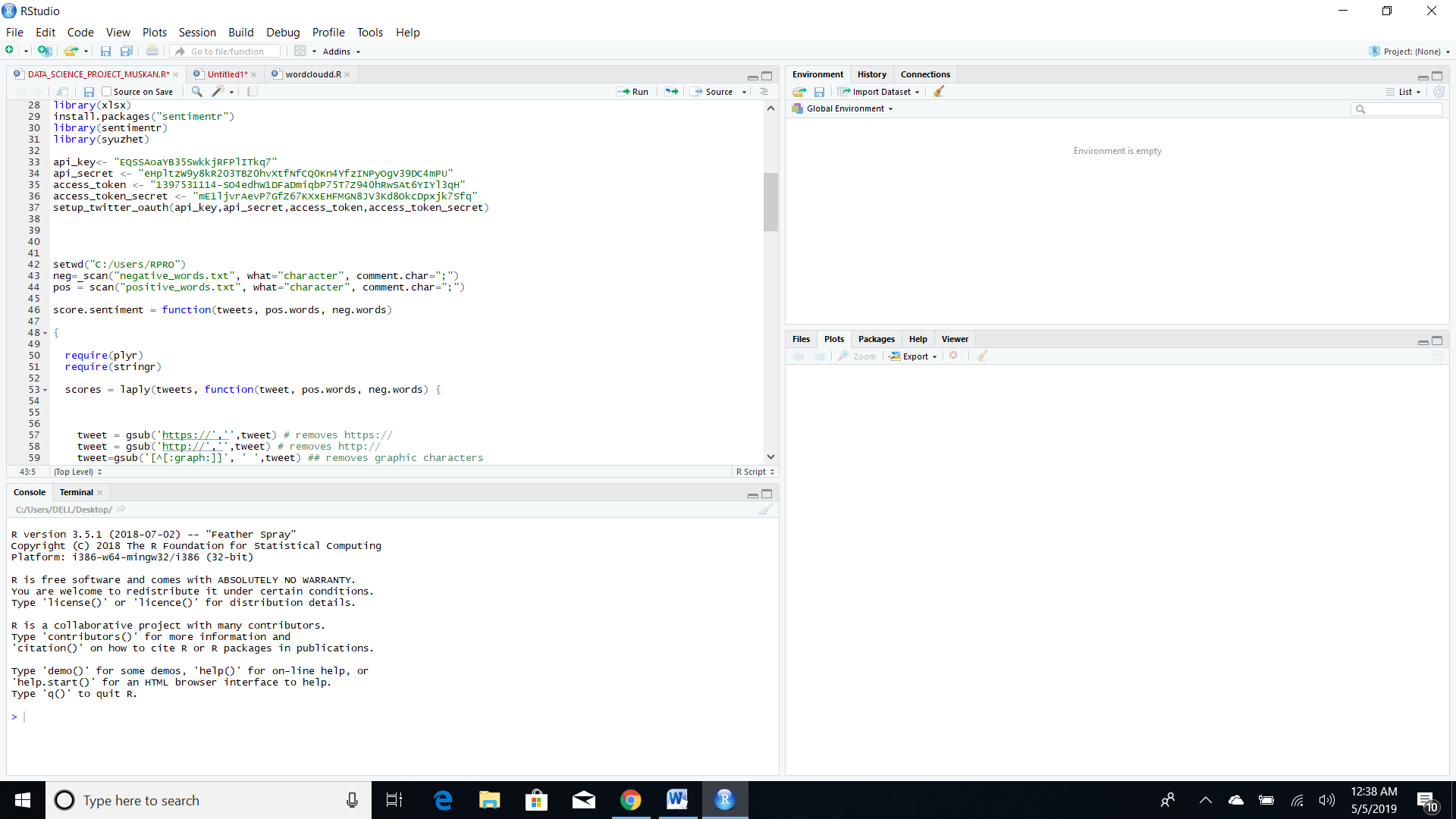
The tweets are cleaned in R by removing:

* Extra punctuation
* Sentences to Lower case
* Redundant Blank spaces and tabs
* Removing URLs

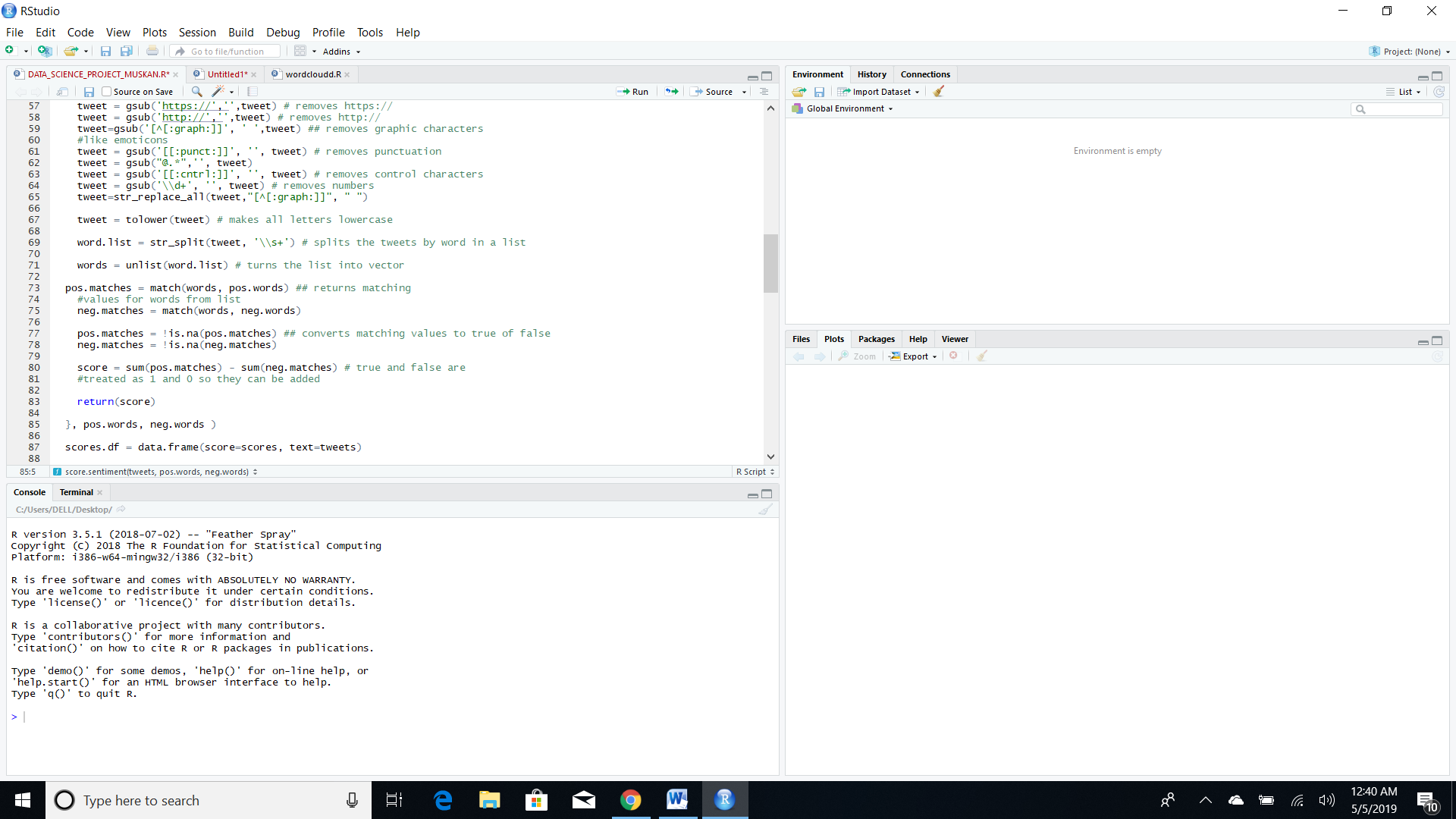


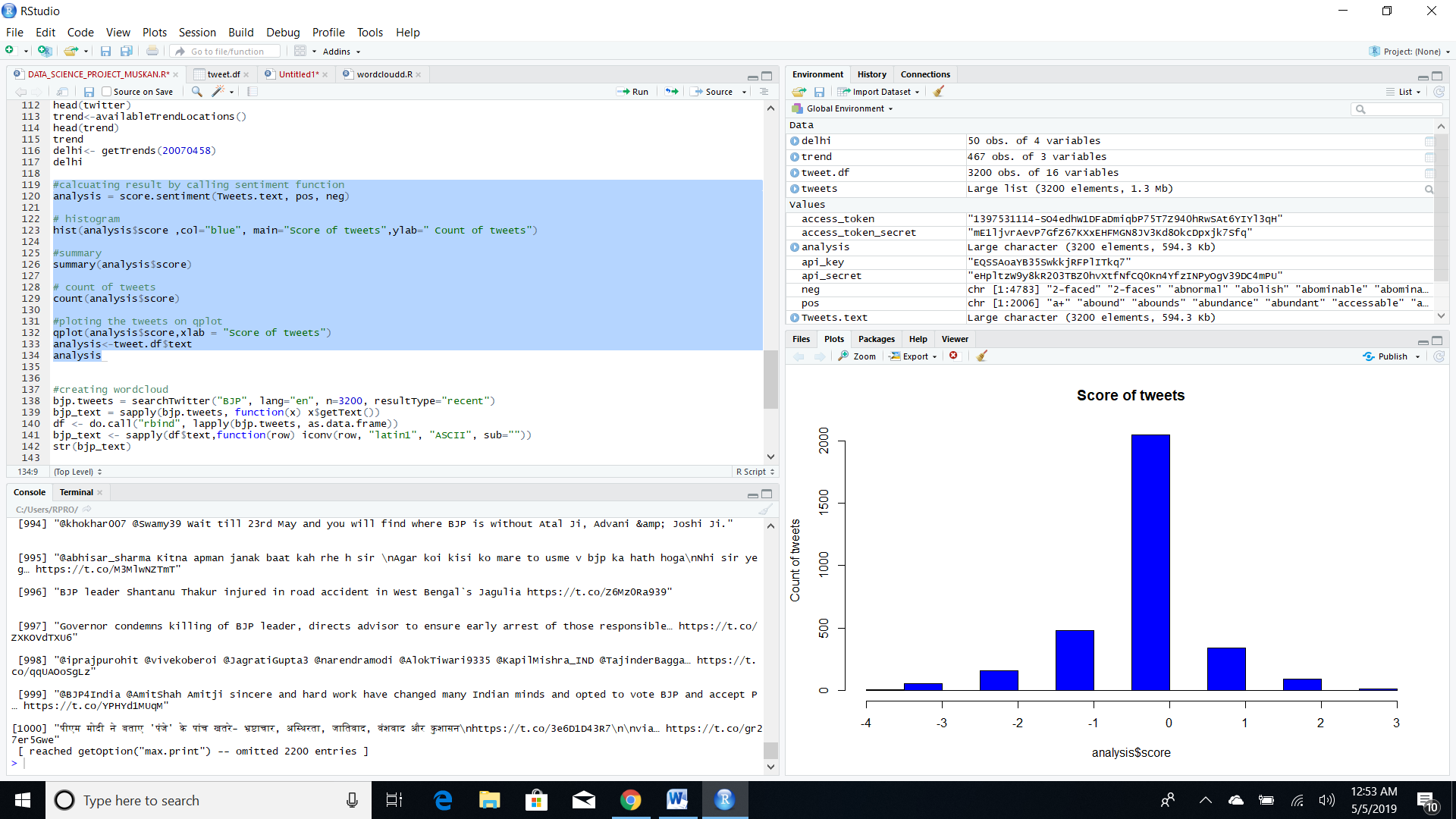
# Step 4 : Loading word Database and Lexical Analysis

A database, containing positive and negative words, is loaded into R. This is used for Lexical Analysis, where the words in the tweets are compared with the words in the database and the sentiment is predicted.



**Lexical Analysis**:​​By comparing uni-grams to the pre-loaded word database, the tweet is assigned sentiment score - positive, negative or neutral and overall score is calculated.

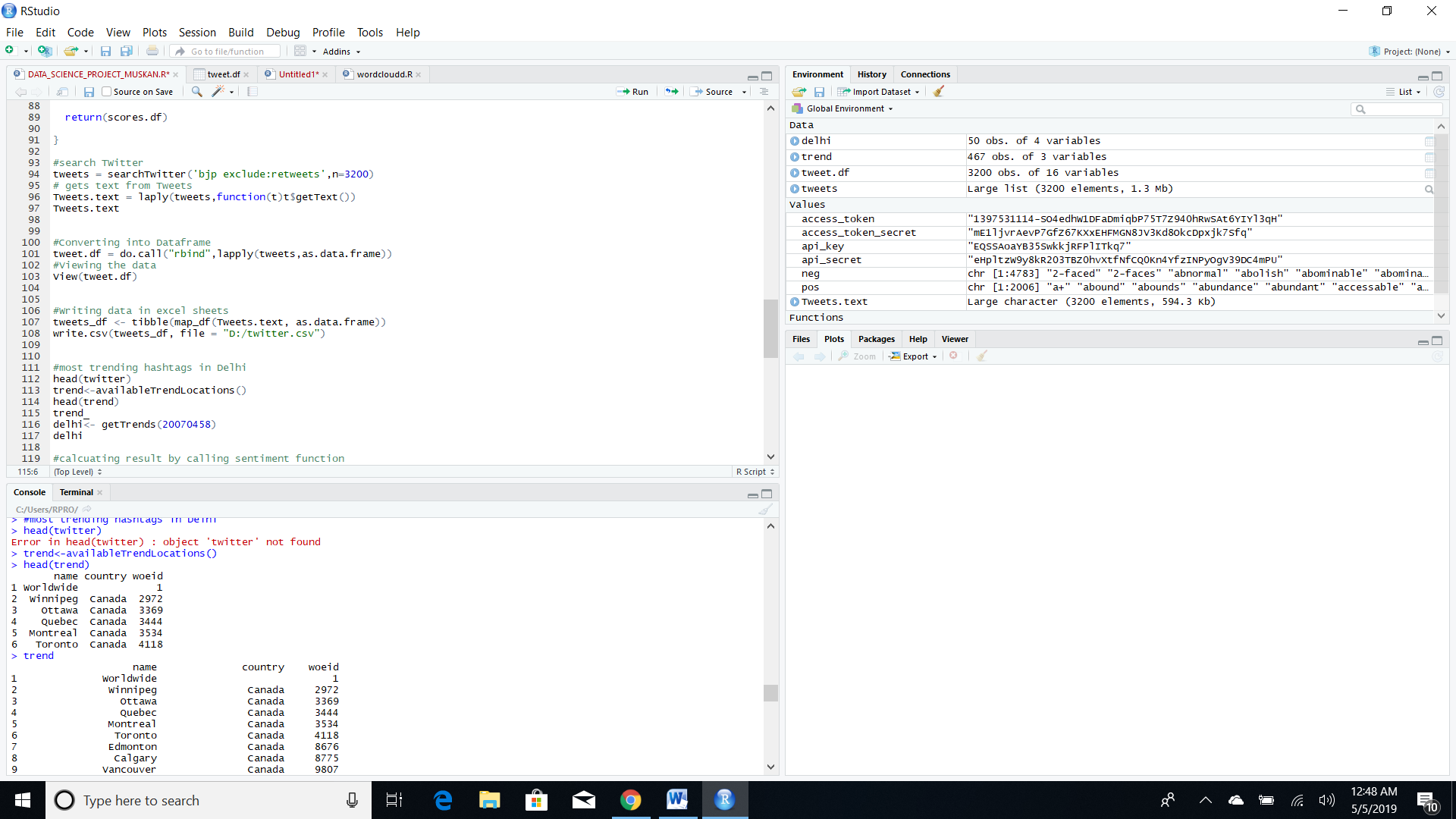


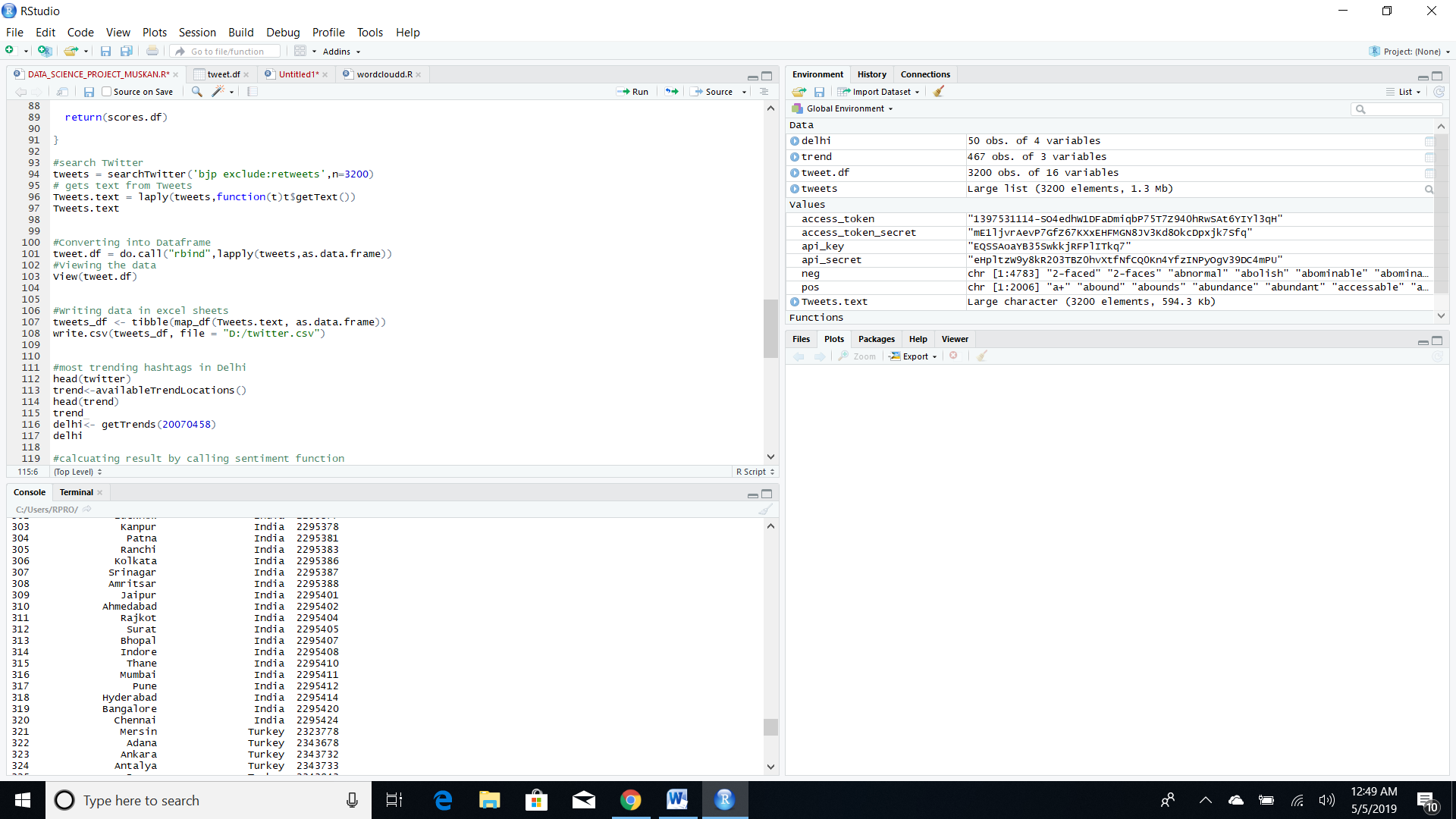


From these visualizations, we get a better understanding of the sentiment associated with tweets. We can infer that most tweets were neutral and the negative tweets are more than positive tweets.

Step 5: Finding Trends of a Country

We explore the number of tweets with #BJP in every country in the world. Then we look at the most trending hashtags in a Country or State.



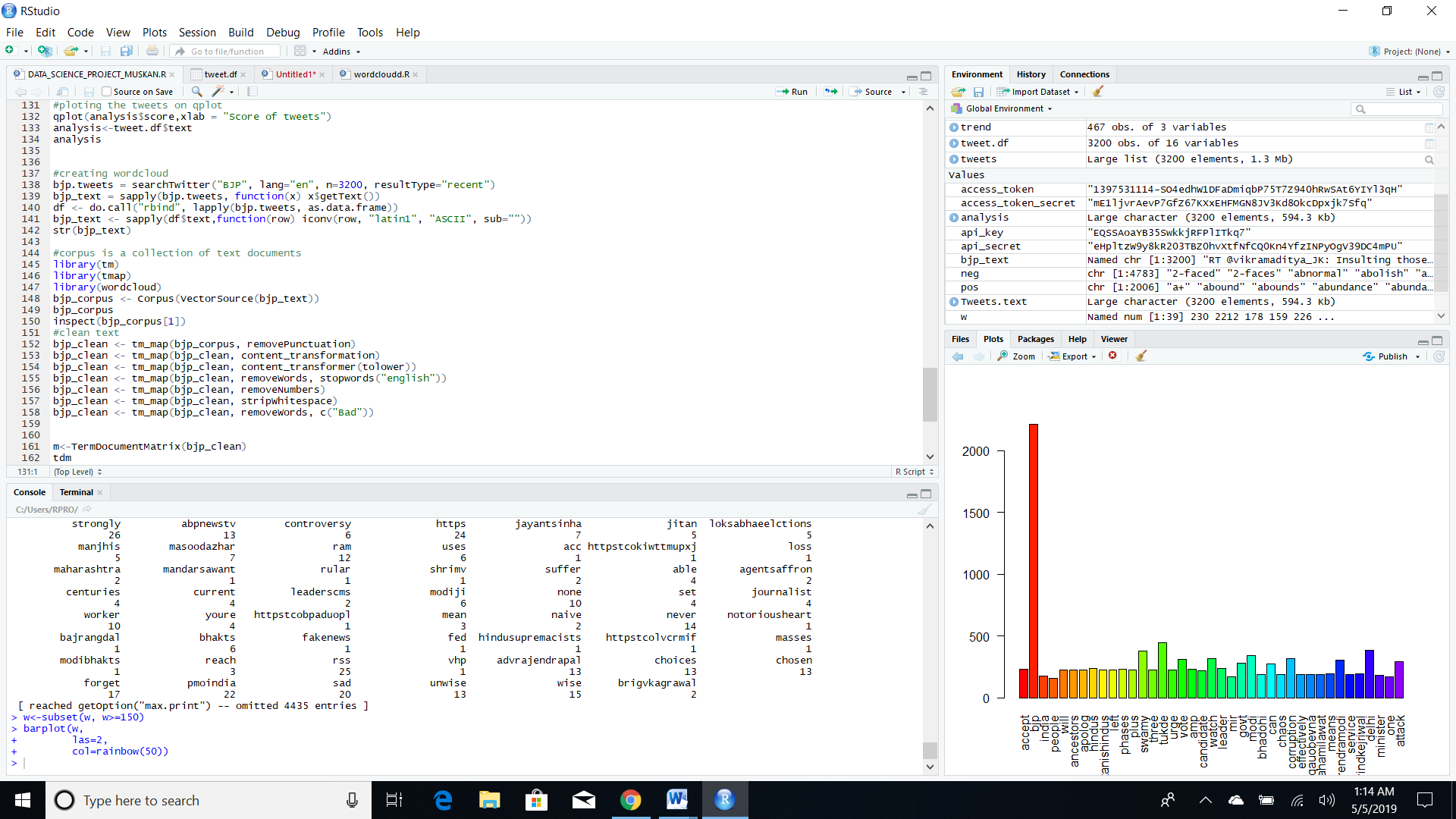


# Most recent hashtags used in Delhi

# 

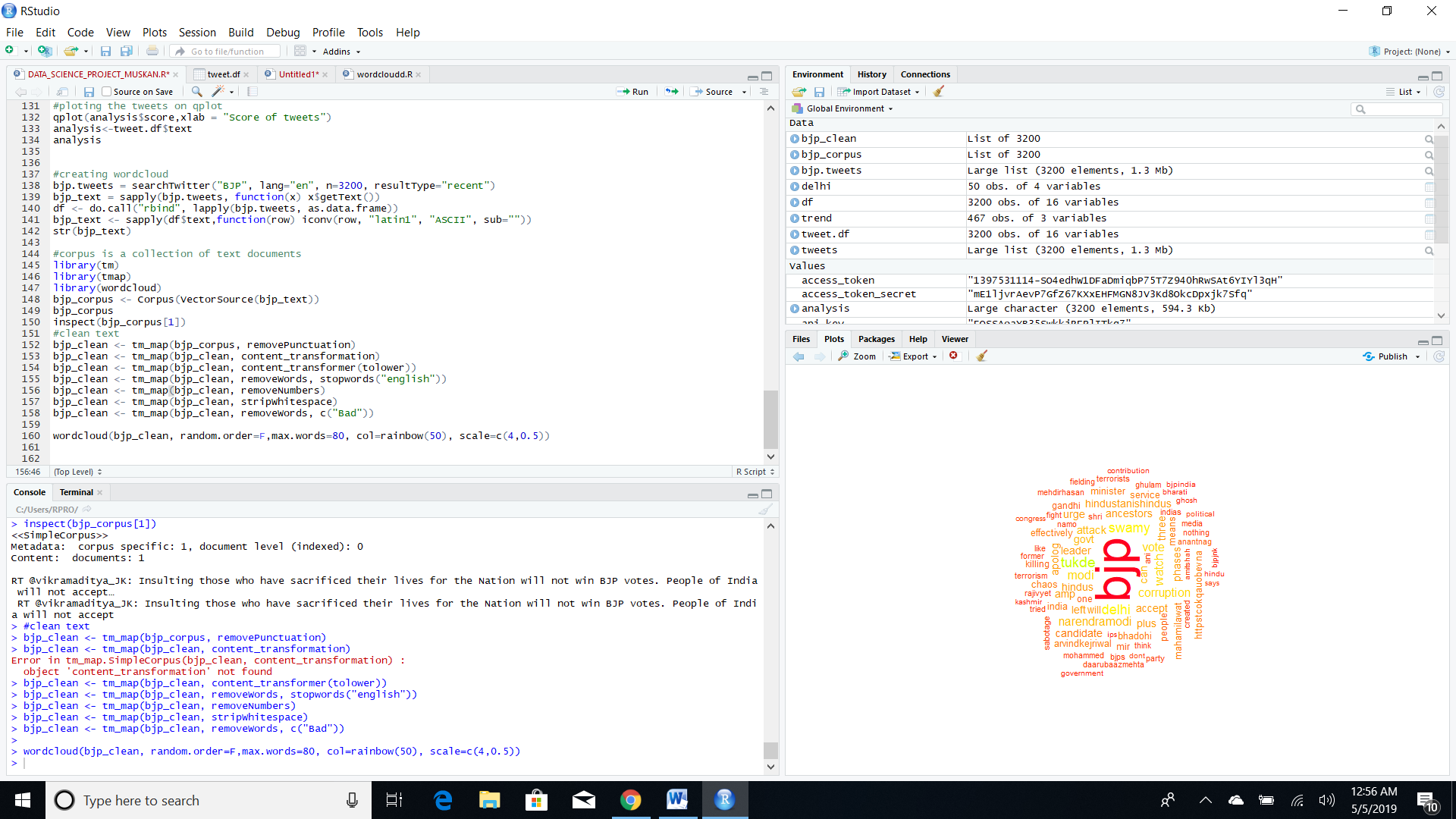
# Step 6: Wordcloud Formation

Text files were then transformed into a R – readable format. The package TM and other text mining packages operate on a format called corpus. Corpus is just a way to store a collection of documents in a R software readable format.



Next the **corpus is cleaned.**Here we will make 4 important transformations.

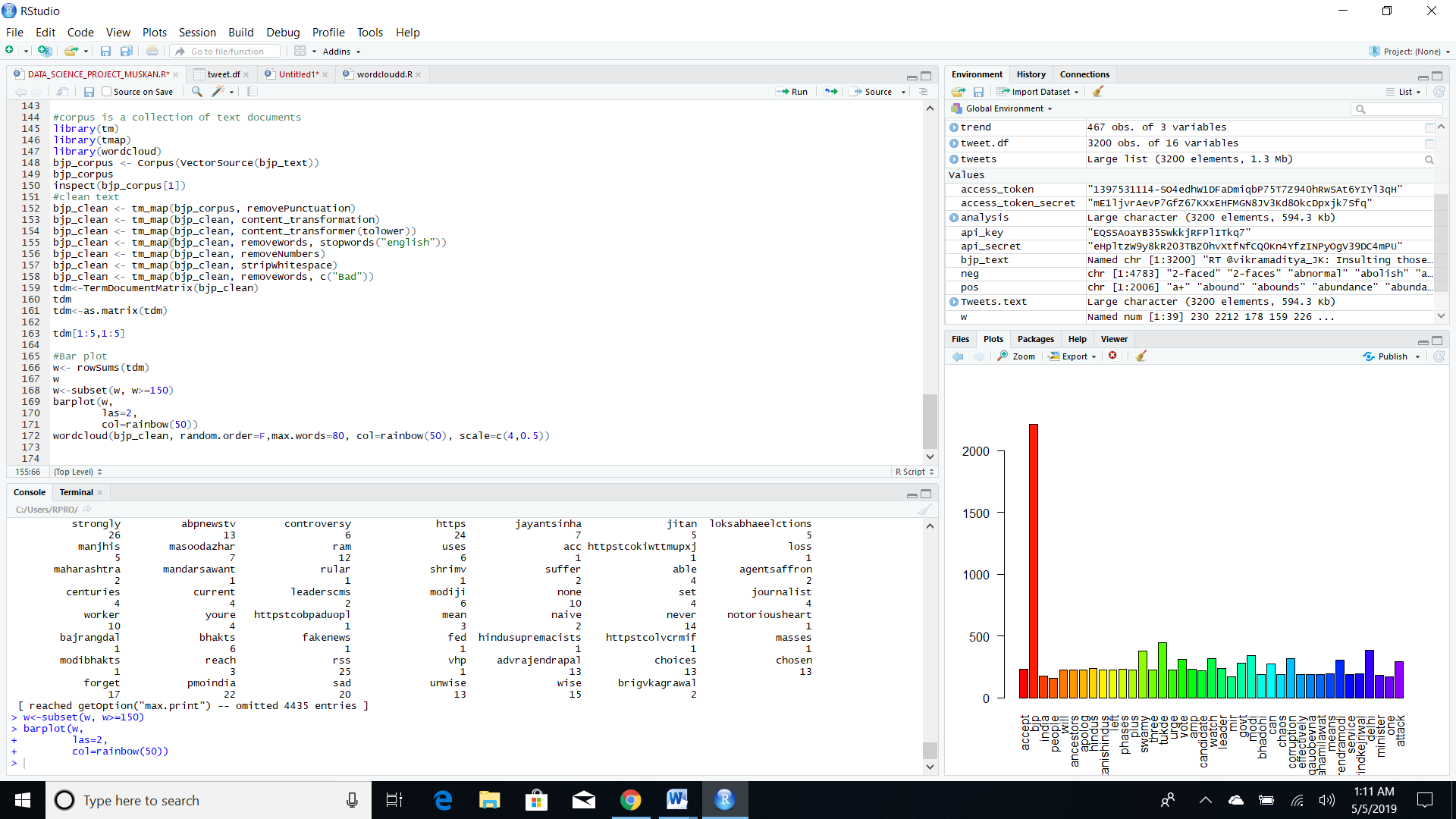
1. Replace symbols like “/” or “@” with a blank space
2. Remove words like “a”, “an”, “the”, “I”, “He” and numbers. This is done to remove any skewness caused by these commonly occurring words.
3. Remove punctuation and finally whitespaces. Note that we are not replacing these with blanks because grammatically they will have an additional blank.
4. Remove words that are repeated way too many times. These words can disrupt the word clouds.

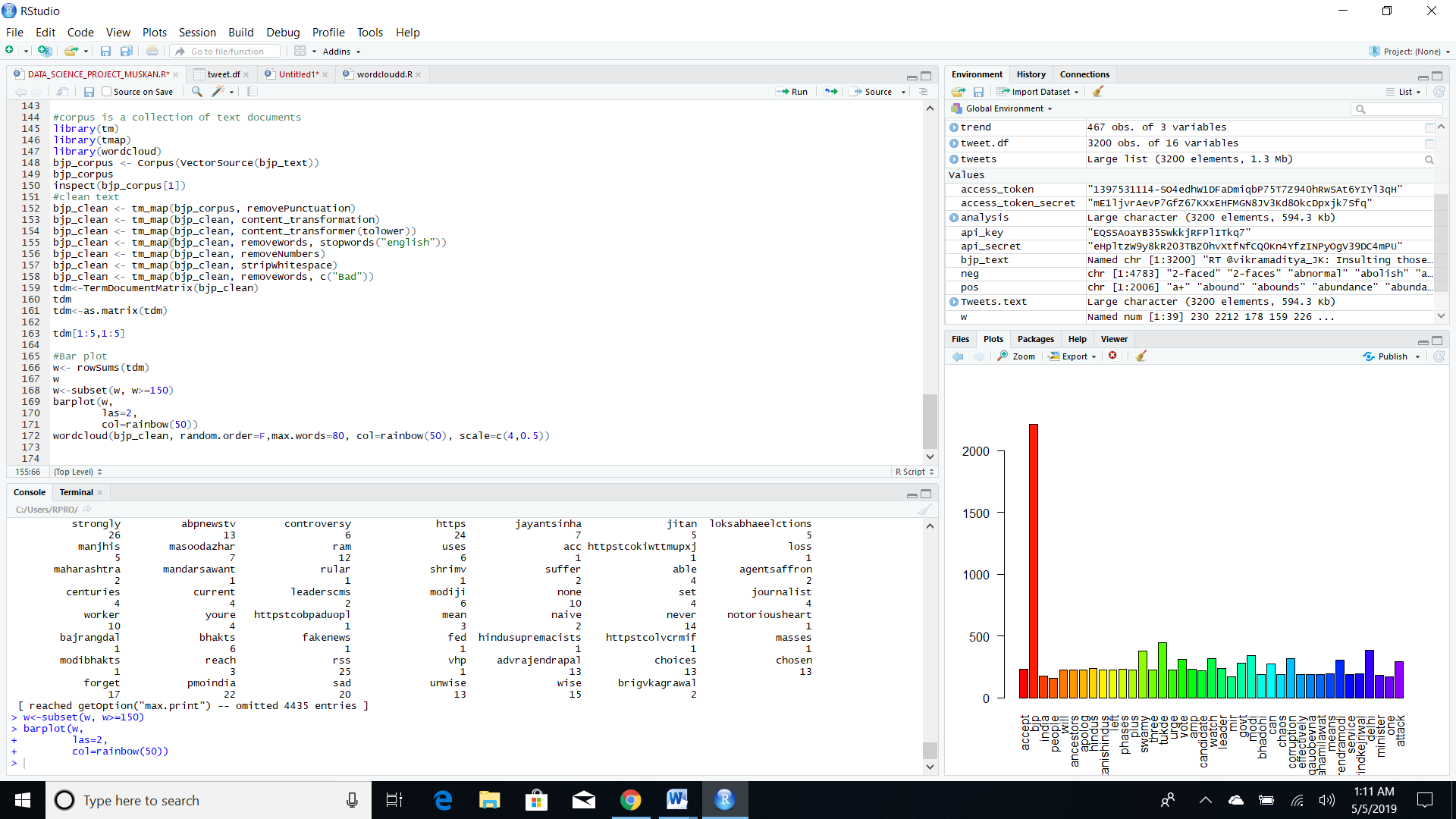


Text data like tweets is unstructured data. So in order to proceed with the analysis, this has to be converted in a structured format. This can achieved by a Term Document Matrix.

# 

I have also visualised the Term Documentation matrix by plotting the row sum, which gives the number of times each word appears. This gives a more detailed insight.





A **word cloud** is a novelty visual representation of text data, typically used to depict keyword metadata (tags) on websites, or to visualize free form text. Tags are usually single words, and the importance of each tag is shown with font size or color.I used the wordcloud function to make wordcloud

**Wordcloud**

