**PROGRAMMING IV PROJECT**

DATA COLLECTION FROM FACEBOOK USING

R PROGRAMMING

Submitted by :

Vikas Sharma : 563/IC/14

Vijay Lakshmi : 562/IC/14

1. Programming Language

1.1 Introduction

R is a programming language and software environment for statistical analysis, graphics representation and reporting. R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team.

R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems like Linux, Windows and Mac.

This programming language was named **R**, based on the first letter of first name of the two R authors (Robert Gentleman and Ross Ihaka), and partly a play on the name of the Bell Labs Language **S**.

1.2 Features of R

As stated earlier, R is a programming language and software environment for statistical analysis, graphics representation and reporting. The following are the important features of R −

* R is a well-developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.
* R has an effective data handling and storage facility,
* R provides a suite of operators for calculations on arrays, lists, vectors and matrices.
* R provides a large, coherent and integrated collection of tools for data analysis.
* R provides graphical facilities for data analysis and display either directly at the computer or printing at the papers.

As a conclusion, R is world’s most widely used statistics programming language. It's the # 1 choice of data scientists and supported by a vibrant and talented community of contributors. R is taught in universities and deployed in mission critical business applications. This tutorial will teach you R programming along with suitable examples in simple and easy steps.

1.3 Applications of R Programming

Around the world, millions of analysts and data researchers use R Programming to take care of their most difficult issues in the fields running from computational science to extensive marketing. R Programming, or R, has turned into the most prevalent language for data science and a fundamental tool for Finance and analytics-driven organizations, for example, Google, Facebook, and LinkedIn.

R is a language and environment for statistical computing and design. It is a GNU venture which is like the S language and environment which was created at Bell Laboratories by John Chambers and Associates. R Programming has brought revolutionary modifications in [Big Data](https://intellipaat.com/big-data-hadoop-training/?utm_campaign=elearningindustry.com&utm_source=%2Fapplications-r-programming-r-eal-world&utm_medium=link) Analytics and other aspects of data analytics and data science. R Programming can be considered as an alternate execution of S. There are some dynamic contrasts. However, much code composed for S runs unaltered under R.

R Programming gives a broad variety of statistical (direct and nonlinear modeling), traditional statistical tests, time-arrangement analysis, grouping, bunching and graphical techniques, and is profoundly extensible. The S language is regularly the vehicle of decision for exploration in statistical methodology, and R gives an Open Source route to cooperation in that action.

R Programming applications compass the universe from hypothetical, computational statistics and the hard sciences, for example, astronomy, chemistry, and genomics to practical applications in business, drug advancement, finance, health care, marketing, medicine and much more. Since R has almost 5,000 packages (libraries of functions) large portions of which are committed to particular applications, you don't need to be an R Programming genius to begin developing your applications.

2. Problem

Data Collection From Facebook Using R Programming.

3. Challenges

In the newer versions of the Facebook API (this post is based on version 2.8) a lot of functions have much higher restrictions. So there are basically two versions to connect to the API: Via an app or via a temporary token. Using the app has the advantage that it is quite comfortable to connect to the API once the app is created. But you won´t all the data available via this method. The connection via the temporary token will give you more data but the token is only valid for 2 hours.

### 4. How to analyze your Facebook friends network with R

The twitteR package, released back in 2010, has long provided the means to access and analyze your Twitter social network data with R. But until recently, there hasn't been anything comparable for the Facebook social network. But now, thanks to Pablo Barbera, there is the RFacebook package which provides a collection of R functions to access data from your Facebook social network.

To use RFacebook, you first need to sign up for a Facebook developer account, which is quick and easy as long as you already have a Facebook profile. JulianHi provides an excellent step-by-step tutorial on getting started with RFacebook, including setting up your Facebook app and generating the authentication token that will be necessary for using the RFacebook functions.

Once you're authenticated, you can use the RFacebook functions to query your friends list and get information about your friends (and their connections with your other friends).

5. Step By Step Implementation

## 

## **5.1 Preparations in R**

## **First we need to install the packages devtools and Rfacebook from github as this is currently the most recent version.**

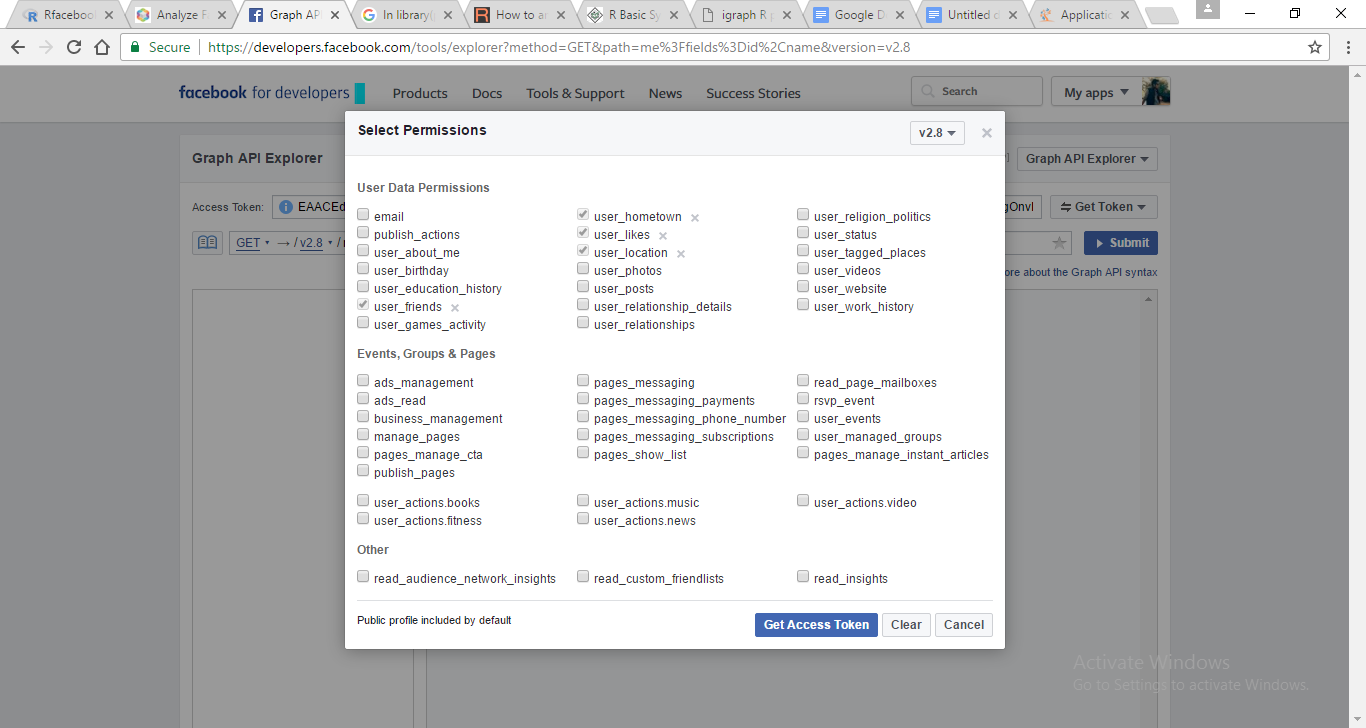
|  |  |  |
| --- | --- | --- |
|  | install.packages("devtools")  library(devtools) |  |

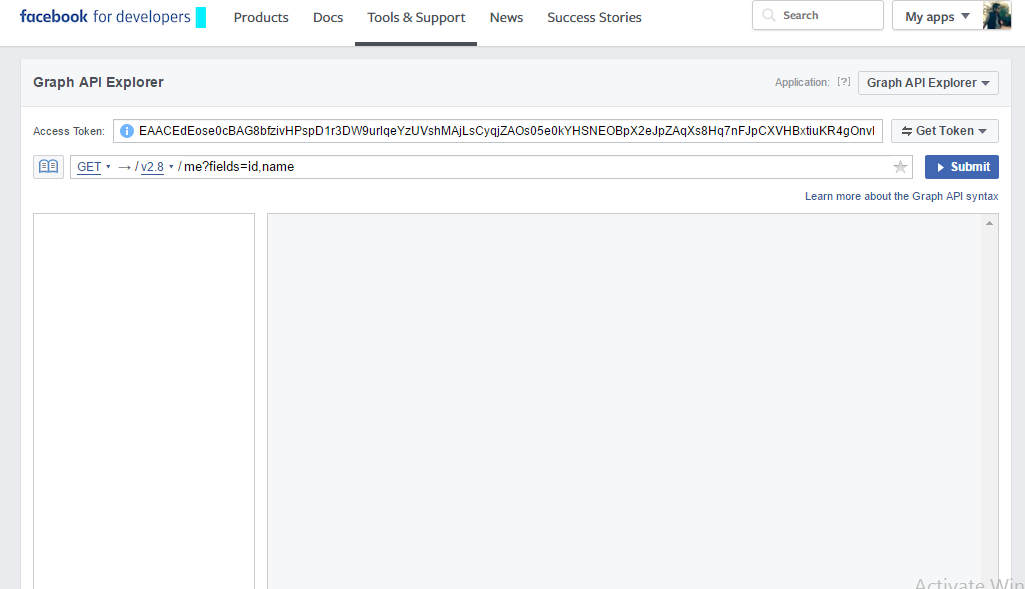
|  |  |
| --- | --- |
|  | install\_github("Rfacebook", "pablobarbera", subdir="Rfacebook")  require (Rfacebook) |

**5.2 Connecting to the Facebook API via Authentication Token:**

The connection via an manually created authentication token has some less restrictions and so you can get more data. However, this token is only valid for 2 hours.

`You can create your token at <https://developers.facebook.com/tools/explorer>





**5.3 Data Collection**

**a) Getlikes**

getLikes retrieves information about a friend’s likes. To retrieve the number of likes for a page, use getUsers with the page IDs

**Usage**

getLikes(user, n = 500, token)

**Arguments**

user A user ID or screen name.

n Maximum number of likes to return for each user.

token Either a temporary access token created at https://developers.facebook.

com/tools/explorer or the OAuth token created with fbOAuth.

**Details**

This function requires the use of an OAuth token with the following permissions: user\_likes, friends\_likes.

**Code**

token<-'EAACEdEose0cBAG8bfzivHPspD1r3DW9urlqeYzUVshMAjLsCyqjZAOs05e0kYHSNEOBpX2eJpZAqXs8Hq7nFJpCXVHBxtiuKR4gOnvEKkHjCGvPecoRcd3nZCEUEPuvoVDVrk59H3DVtY60X1Pu06QIR47K0pIRWe2t5ASFGUUVg1CvhoP4'

my\_likes <- getLikes(user="me",n=50, token)



**b) GEtfriends**

**Description**

getFriends retrieves information about the user’s friends.

**Usage**

getFriends(token, simplify = FALSE)

**Arguments**

token Either a temporary access token created at https://developers.facebook.

com/tools/explorer or the OAuth token created with fbOAuth.

simplify If TRUE, function will return only name and id for each friend. If FALSE, it will

return additional information from their profiles: gender, birthday, location,

hometown,relationship status and profile picture.

**Details**

This function requires the use of a OAuth token with extended permissions. After the introduction of version 2.0 of the Graph API, only friends who are using the application that you used to generate the token to query the API will be returned.

**Code**

## Not run:

## Copy and paste token created at FB Graph API Explorer

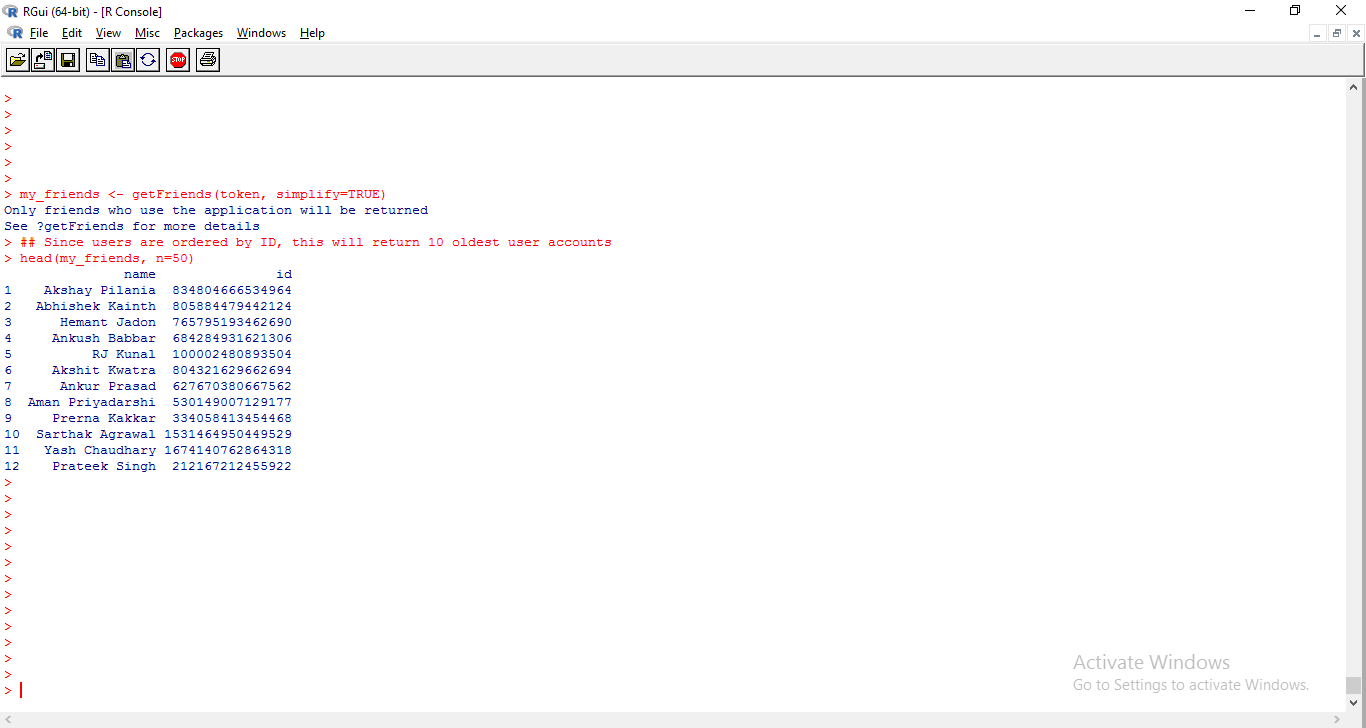
token<-"EAACEdEose0cBAFkDm7XbZAYeZBJsHKKxrnJ25rgnTiaDUBlAZAF2itRsRoHROcNL57oq8D967Lj5k0ZCRxLqEn5EmcvTkxdZAdaZBR1sqQ0mNXAantdVhVWZCzF95JugbzQrkBht9QozwbtRwZAolVnOF8h7WydAdaowg4u9tJZAoTIvNnPM8FwGA"

my\_friends <- getFriends(token, simplify=TRUE)

## Since users are ordered by ID, this will return 10 oldest user accounts

head(my\_friends, n=50)

## End(Not run)



**c) Getnetwork**

**Description**

getNetwork retrieves the list of mutual friendships and returns the adjacency matrix or edge list for the network representing the neighborhood of the authenticated user.

**Usage**

getNetwork(token, format = "edgelist", verbose = TRUE)

**Arguments**

Token Either a temporary access token created at https://developers.facebook.

com/tools/explorer or the OAuth token created with fbOAuth.

format Either "edgelist" for a list of edges in the network or "adj.matrix" for an

adjacenty matrix of dimensions (n x n), with n being the number of friends, and

0 or 1 indicating whether friend i is also friends with friend j.

verbose logical,default is TRUE, which will print additional information on the console.

**Details**

This function requires the use of an OAuth token with extended permission. After the introduction of version 2.0 of the Graph API, only friends who are using the application will be returned

**Code**

I used the code below to query my Facebook friends, and used the igraph package to draw my social network:

require(Rfacebook)

token<-'EAACEdEose0cBAFkDm7XbZAYeZBJsHKKxrnJ25rgnTiaDUBlAZAF2itRsRoHROcNL57oq8D967Lj5k0ZCRxLqEn5EmcvTkxdZAdaZBR1sqQ0mNXAantdVhVWZCzF95JugbzQrkBht9QozwbtRwZAolVnOF8h7WydAdaowg4u9tJZAoTIvNnPM8FwGA'

me <- getUsers("me", token)

my\_friends <- getFriends(token, simplify=TRUE)

my\_friends\_info <- getUsers(my\_friends$id, token, private\_info=TRUE)

my\_network <- getNetwork(token, format="adj.matrix")

singletons <- rowSums(my\_network)==0 # friends who are friends with me alone

require(igraph)

my\_graph <- graph.adjacency(my\_network[!singletons,!singletons])

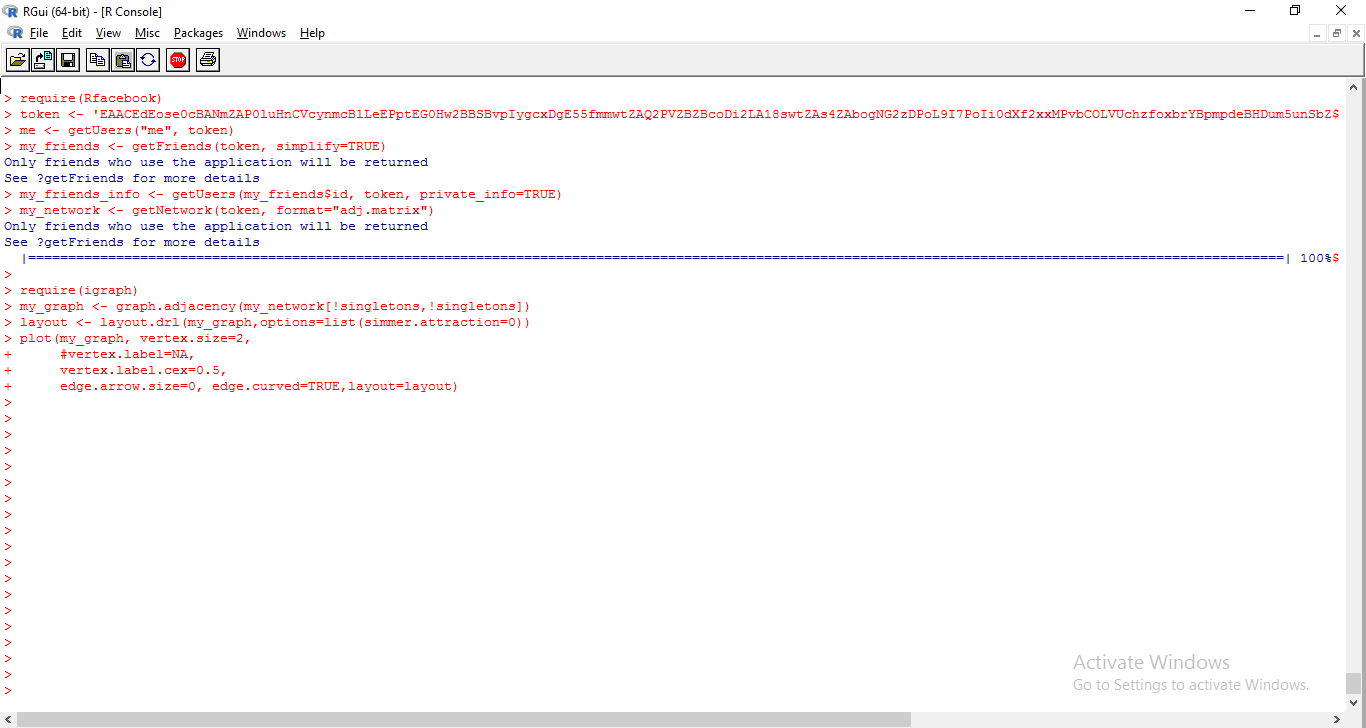
layout <- layout.drl(my\_graph,options=list(simmer.attraction=0))

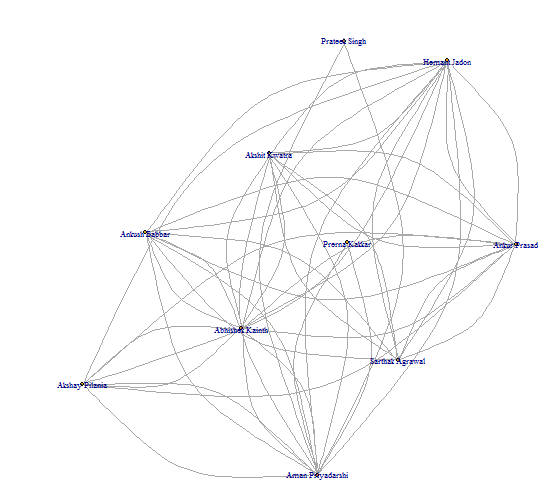
plot(my\_graph, vertex.size=2,

#vertex.label=NA,

vertex.label.cex=0.5,

edge.arrow.size=0, edge.curved=TRUE,layout=layout)



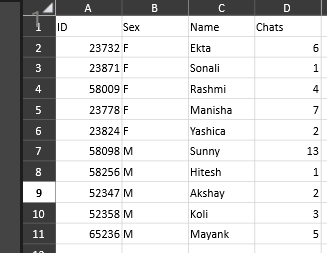


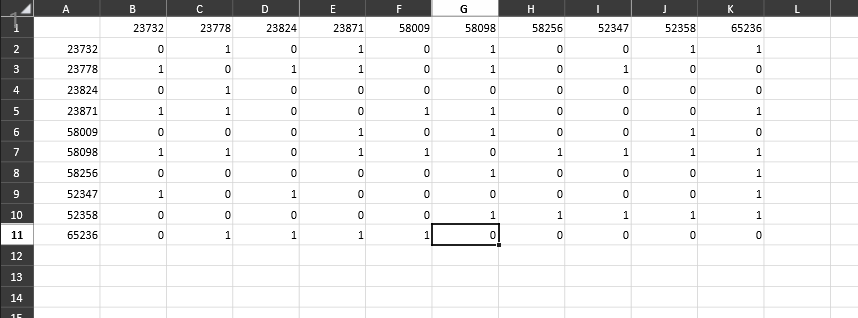
6.Network Analysis of Data from MS Excel

Step by step implementation

1)Enter data in ms exel

a)Data about attributes



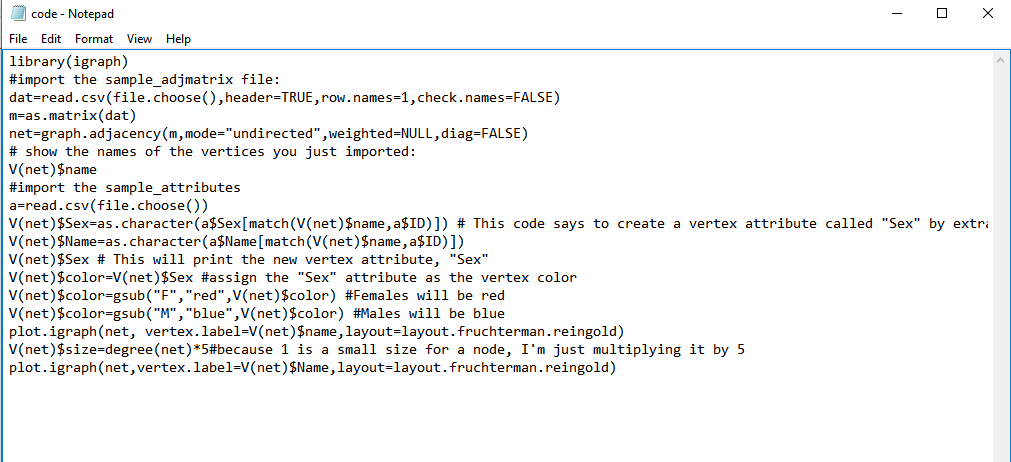
b)Adjacency matrix data 

2) Convert .xlsx file to .csv file

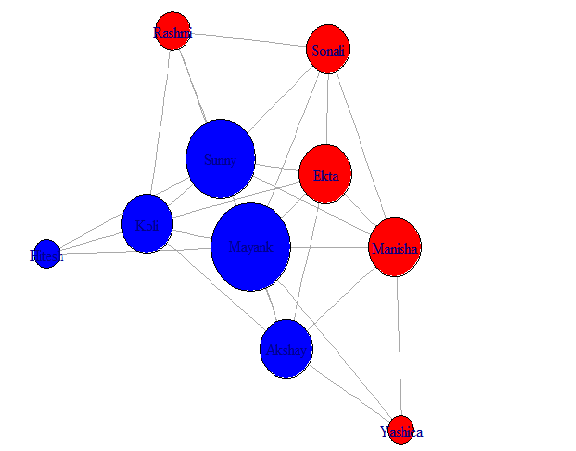
we have used below link to convert

<https://convertio.co/xlsx-csv/>

3) Code for importing .csv files to r studio as well as creating a network



4)Output



5)Description of output

This graph depicts the connections of friends according to data of the exel file.

Red nodes define female friends and blue node defines male.

Size of node depends upon degree(connections) of each node.

6. Shortcomings

* The connection via an manually created authentication token has some less restrictions and so you can get more data. However, this token is only valid for 2 hours.
* We can only collect public data or data related to user using Rprogramming.

7. Conclusion

One can find this small piece of analysis bot interesting and insightful. It helps one to get a summary of everything. One can go through the user information of his intire friend list in less than 5 minutes. One can use this data to visualize his friends on a graph and see various clusters of population, one can make a word cloud from the frequency of texts. One can do something cool things like define the distance between nodes basis interactions on facebook and see which are the closet people to you as per facebook and much more things but since facebook is increasing it’s securities and now does not shares it’s all data, therefore one has to take care of not violating any rules.

8. Bibliography

* [**https://cran.r-project.org/web/packages/Rfacebook/Rfacebook.pdf**](https://cran.r-project.org/web/packages/Rfacebook/Rfacebook.pdf)
* [**https://developers.facebook.com/tools/explorer?method=GET&path=me%3Ffields%3Did%2Cname&version=v2.8**](https://developers.facebook.com/tools/explorer?method=GET&path=me%3Ffields%3Did%2Cname&version=v2.8)
* [**http://blog.revolutionanalytics.com/2013/11/how-to-analyze-you-facebook-friends-network-with-r.html**](http://blog.revolutionanalytics.com/2013/11/how-to-analyze-you-facebook-friends-network-with-r.html)
* [**http://thinktostart.com/analyzing-facebook-with-r/**](http://thinktostart.com/analyzing-facebook-with-r/)
* **<https://www.tutorialspoint.com/r/r_basic_syntax.htm>**
* **<https://elearningindustry.com/applications-r-programming-r-eal-world>**