

Assignment

For your assignment, you will work on two graphs which were created out of the Twitter data you used in the lab session.

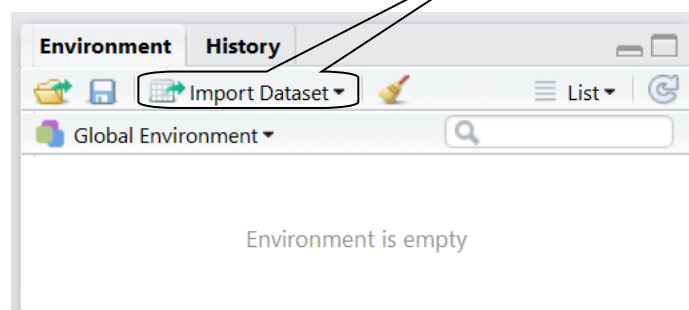
1st graph: Retweet Network

- A “retweet” is the situation where a tweet by user A is tweeted out again by a user B. This is indicated by a beginning “RT” string in the tweet.
- To form the retweet network
 - For each tweet of A which is retweeted by B,
 - We placed a directed edge from B to A.
 - If there is already an edge from B to A, we incremented the weight of the edge by 1; otherwise, the weight of the edge is set to 1.
- **The data is in retweet_relations.csv (159 MB, 7,224,942 rows)**
 - Each row of the data is a relation between two users (each user is indicated with an integer), for instance:

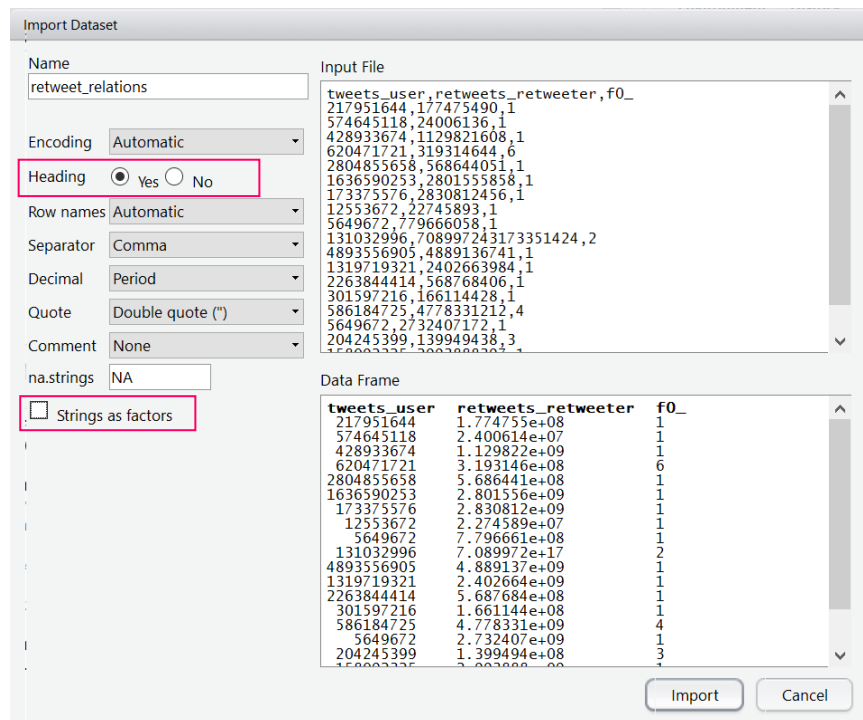
959786971	145049088	1
-----------	-----------	---


means that there is an edge from user 959786971 to user 145049088 of weight 1.

- You can open the document with Excel to examine the content. However, bear in mind that Excel will load only the first 1,000,000 rows of the data.
- To load the data into R, click on the “Import Dataset” icon and select “From Local File...”



Then, ensure that the “Heading” and “Strings as factors” are set as below:



Finally, click , and you will find the data loaded as a data frame called “retweet_relations”.

- To rename the variable to some shorter name, say “mydata”, first create a copy, and then remove the old one


```
> mydata <- retweet_relations
> remove(retweet_relations)
```

2nd graph: Mention Network

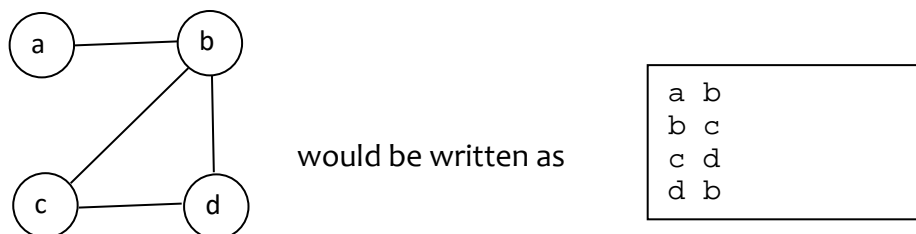
- A “mention” is the situation where a tweet by user B mentions a user A. This would be indicated by a beginning “@A” string in the tweet.
- To form the mention network
 - For each tweet of A which is mentioned by B,
 - We placed a directed edge from B to A,
 - If there is already an edge from B to A, we incremented the weight of the edge by 1; otherwise, the weight of the edge is set to 1.
- The data is in mention_relations.csv (339 MB, 15,432,539 rows)

What you need to do

1. Write an R function which finds the average in-degree and out-degree of each of the two graphs.
2. Write an R function which finds the total number of nodes for each of the two graphs.
3. Write an R function which performs the following task:
 - Create a new graph G out of the Retweet Network, N , such that
 - There is an (undirected) edge in G between two users A and B if and only if the edges $A \rightarrow B$ and $B \rightarrow A$ are in N .
 - That is, an edge is in G if and only if A has retweeted some tweets from B and B retweeted some tweets from A .
 - You will find that your function will take many hours to run. In order to enable it to complete, you will have to run with fewer rows. You can obtain only the first n rows out of a data frame by the command `head(., n)`. For example


```
> mypartialdata <- head(mydata, 10000)
```

 will take the first 10000 and place it in the variable "mypartialdata".
 - After you complete the graph, visualize it with the pals system (<http://104.155.190.239:3001/>). To do so, prepare your graph as a text file which lists out all the edges. For example, the graph



To use the pals system, select “upload graph”, followed by [Browse]. Then, choose your file. Finally, press [Submit Me].

Pals Graph Communities Visualizer

built in graph ☐ upload graph ☒ No file selected.

compute communities ☒ upload communities ☐ No file selected.

- The pals system will only accept graphs of up to around 500 vertices. Hence, you will need to trim your subgraph if you have much more than 500 vertices.

Submitting your answers

1. Prepare the following plain text files:

- a. **“program1.R”** — this file should contain your program for the 1st problem.
- b. **“program2.R”** — this file should contain your program for the 2nd problem.
- c. **“program3.R”** — this file should contain your program for the 3rd problem.
- d. **“graph.txt”** — the (graph) file which you uploaded to the pals system.
- e. **“answers.txt”** — this file should contain the following content:

```
Retweet
<average in-degree of retweet network>
<average out-degree of retweet network>
<number of nodes in retweet network>
Mention
<average in-degree of mention network>
<average out-degree of mention network>
<number of nodes in mention network>
Subgraph
<number of rows after trimming with head(,,) (7,224,942 if no trimming)>
<number of rows in graph.txt above>
```





Replace each <description> line above with the actual value you obtained or used.


Also, prepare a PNG image called **“network.png”** — this file should contain a screenshot of the graph you obtained for the 3rd problem as generated by pals.

2. Create a zip archive to put your answers.

- a. Right click on your desktop and select “New→Compressed (Zipped) Folder”. This will create a zip file called “New Compressed (zipped) Folder.zip”.
- b. Change the name of the zip file to your IVLE login name. For example, if your login name is “nusri16000”, you should change the zip file’s name to “nusri16000.zip”.
- c. Drag add all your files (**program1.R**, **program2.R**, **program3.R**, **graph.txt**, **answers.txt**, **network.png**) into the zip file to add them to it. (Open the zip file to double-check that all your files are inside.)

3. You should submit your answers through IVLE. First, login to IVLE.
4. After you login to IVLE, go into the folder “FILES/Lab-Assignment”.

	Name	Size
<input type="checkbox"/>	 Lecture Notes	12.62 MB
<input type="checkbox"/>	 Student Submission	0 Bytes
<input type="checkbox"/>	 Assignments	589.10 MB
<input type="checkbox"/>	 Lab-Assignment	2.35 KB

5. Press the  Upload button to upload your documents.
6. You will need to upload your zip file before 10AM, 18th July (Monday). *(If you failed to meet the deadline, upload your zip file to the folder “Student Submission” instead, as soon as possible.)*

Grading

1. Grading will be performed on the lab session on Monday.
2. You will be graded according to the marking scheme in the next page.

Computational Thinking: Algorithms for Big Data Community Detection Basic
(RI2001A)

Assignment Evaluation

--

(Total: 50 marks)

Student Name: _____

Average in-degree:

Algorithm Correctness (5 marks)	Algorithm Efficiency (2 marks)

Average out-degree:

Algorithm Correctness (5 marks)	Algorithm Efficiency (2 marks)

Number of nodes:

Algorithm Correctness (5 marks)	Algorithm Efficiency (2 marks)

Subgraph generation:

Algorithm Correctness (5 marks)	Algorithm Efficiency (2 marks)
Size of partial graph (8 marks)	Pals output (4 marks)

Presentation:

Understanding of algorithm (5 marks)	Handling of further questions (5 marks)

Evaluated by: _____