COURSEWORK - 2

Name: Mohammed Ashiq Ariff

# The dataset and tools used:

The chosen Data set is Wikipedia Vote Network available at

<http://snap.stanford.edu/data/wiki-Vote.html>

The dataset has a directed network made from administrator elections with total of 2794 elections.

This coursework is done using R. And the libraries used are igraph

|  |  |
| --- | --- |
| Nodes | 7115 |
| Edges | 103689 |
| Nodes in largest WCC | 7066 (0.993) |
| Edges in largest WCC | 103663 (1.000) |
| Nodes in largest SCC | 1300 (0.183) |
| Edges in largest SCC | 39456 (0.381) |
| Average clustering coefficient | 0.1409 |
| Number of triangles | 608389 |
| Fraction of closed triangles | 0.04564 |
| Diameter (longest shortest path) | 7 |
| 90-percentile effective diameter | 3.8 |

# Primary Data Analysis and Building a graph

The data is in the format of table of fromNodes to toNodes. With each having a node id. To build the graph, I used the read.table function on a text file. And then built a graph using graph function from the igraph library.



The Summary of the graph

summary**(**graph**)**

IGRAPH DN**--** 7115 103689 **--**

**+** attr**:** name **(**v**/**c**)**

7115 nodes and 103689 Edges

## Few other Primary Analysis

#Is it connected

**>** is.connected**(**graph**)**

**[**1**]** **FALSE**

#How many components

**>** no.clusters**(**graph**)**

**[**1**]** 24

#How big are the components

**>** table**(**clusters**(**graph**)$**csize**)**

2 3 7066

20 3 1

# Degree Distribution

## In- Degree

**>** max**(**degree**(**graph, mode**=**"in"**))**

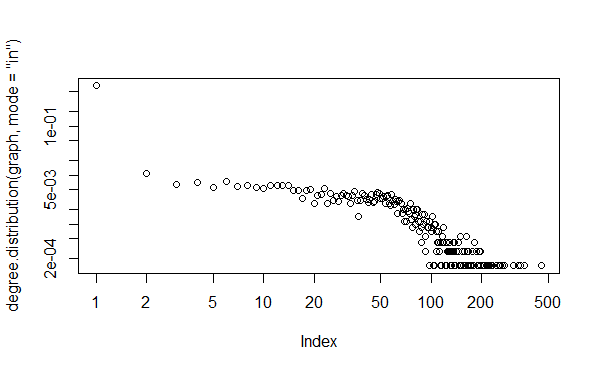
**[**1**]** 457

Plotting the In – Degree Distribution

# In-degree distribution

plot**(**degree.distribution**(**graph, mode**=**"in"**)**, log**=**"xy"**)**

**Plot of the In Degree Distribution**



## Out- Degree

#Calcuate Maximum Vertex out degree

**>** max**(**degree**(**graph, mode**=**"out"**))**

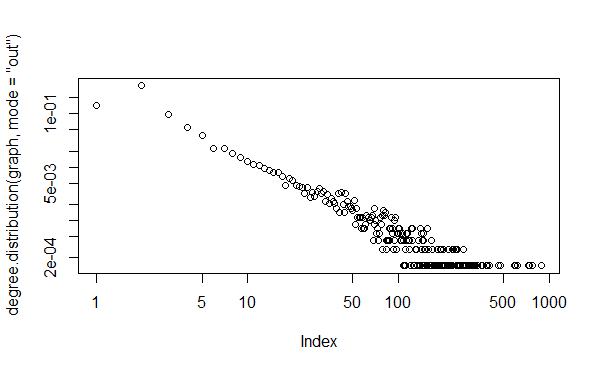
**[**1**]** 893

Plotting the Out – Degree Distribution

# In-degree distribution

**>** plot**(**degree.distribution**(**graph, mode**=**"out"**)**, log**=**"xy"**)**

**Plot of the Out Degree Distribution**



## Average Degree of the network

This states the average connections each node has with other nodes

mean(degree(graph))

[1] 29.14659

# Community Detection

## Finding the graph density

**>** graph.density**(**graph**)**

**[**1**]** 0.002048538

## Finding the transitivity or clustering coefficient

**>** transitivity**(**graph**)**

**[**1**]** 0.1254791

## Finding The Modularity and Plotting it

Used Fast greedy Method to find community detection and community related attributes

#greedy method

fcomm **<-** fastgreedy.community**(**as.undirected**(**graph**))**

#to find membership attribute

memb **<-** cutat**(**fcomm, steps**=**which.max**(**fcomm**$**modularity**))**

#layout for the graph

lay **<-** layout.drl**(**graph **=** graph**)**

empt **<-** graph.empty**(**n**=**vcount**(**graph**))**

colbar **<-** rainbow**(**5**)**

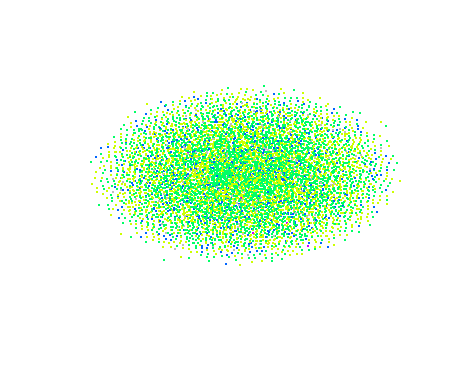
col **<-** colbar**[**memb**+**1**]**

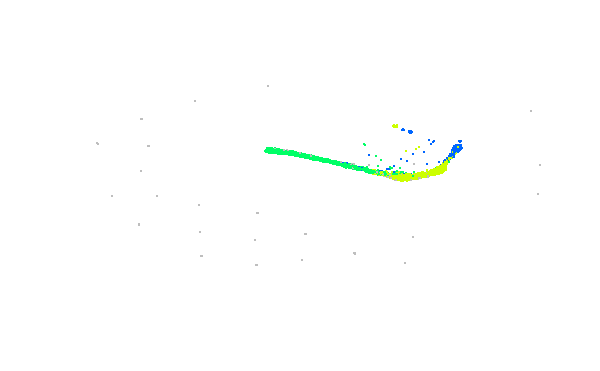
col**[**is.na**(**col**)]** **<-** "gray"

#Plotting the community

plot**(**empt, layout**=**lay, vertex.size**=**1,vertex.label**=NA**, asp**=FALSE**,vertex.color**=**col,vertex.frame.color**=**col**)**

**Before Layouting**



**After Layouting for better split of communities**

# ANALYSIS

## MOST INFLUENTIAL USER IN NETWORK

To find this, I had to remove all the unwanted vertices with a degree less than 200. Thus giving a very small subset from the over all network. This is done In R by

#Analysis - Who is highly influential in the network

sub\_graph **<-** V**(**graph**)[(**degree**(**graph**)/**10**)<**20**]**

The next part is to delete the selected vertices which are with degree less than 200. This is done by

new\_graph **<-** delete.vertices**(**graph **=** graph, sub\_graph**)**

After the deleting the vertices, to decide the size of the vertex, we calculate the degree/10 to give a variable size depending on its importance.

V**(**new\_graph**)$**size**<-**degree**(**new\_graph**)/**10

This is to adjust the margins

par**(**mai**=**c**(**0,0,1,0**))**

Plotting with variable font size of the label and variable size of the node.

plot**(**new\_graph, layout**=**layout.fruchterman.reingold,vertex.label.font**=**2, vertex.label.cex**=**V**(**new\_graph**)$**size**/**37,vertex.color**=**col,vertex.frame.color**=**col**)**

**The figure showing the influential users on the network**

The bigger the node size is, the more influential they are. In this case, they are the leaders in the election result. UserID 2565 and 1549 are the top winners.

