Social Information and Networking Digital Assignment 1

Name: Om Ashish Mishra

Registration Number: 16BCE0789

Slot: A2+TA2

The Questions:

Q1. Using the following visualization softwares for social network

a. R Tools for Social Network Analysis or Gephi

b. Social Networks Visualiser (SocNetV)

c. Pajek Visualize your own social network from Facebook. (5 Marks)

Q2. Read through the following papers and summaries the work carried out in those papers.

a. Say It with Colors: Language-Independent Gender Classification on Twitter

b. TUCAN: Twitter User Centric ANalyzer.

c. A Case Study in Text Mining: Interpreting Twitter Data From World Cup Tweets

PS: The papers have been uploaded for reference in schoology. (5 marks)

The Answers:

1.

a.

data = Edge

```
> data = read.csv(file.choose(),header=T)
> data
       source      target weight
1       C-3PO       R2-D2     17
2        LUKE       R2-D2     13
3     OBI-WAN       R2-D2      6
4        LEIA       R2-D2      5
```

```
5          HAN        R2-D2      5
6    CHEWBACCA        R2-D2      3
7      DODONNA        R2-D2      1
8    CHEWBACCA      OBI-WAN      7
9        C-3PO    CHEWBACCA      5
10   CHEWBACCA         LUKE     16
11   CHEWBACCA          HAN     19
12   CHEWBACCA         LEIA     11
13   CHEWBACCA DARTH VADER       1
14   CHEWBACCA      DODONNA      1
15       CAMIE         LUKE      2
16       BIGGS        CAMIE      2
17       BIGGS         LUKE      4
18 DARTH VADER         LEIA      1
19        BERU         LUKE      3
20        BERU         OWEN      3
21        BERU        C-3PO      2
22        LUKE         OWEN      3
23       C-3PO         LUKE     18
24       C-3PO         OWEN      2
25       C-3PO         LEIA      6
26        LEIA         LUKE     17
27        BERU         LEIA      1
28        LUKE      OBI-WAN     19
29       C-3PO      OBI-WAN      6
30        LEIA      OBI-WAN      1
31       MOTTI       TARKIN      2
32 DARTH VADER        MOTTI      1
33 DARTH VADER       TARKIN      7
34         HAN      OBI-WAN      9
35         HAN         LUKE     26
36      GREEDO          HAN      1
37         HAN        JABBA      1
38       C-3PO          HAN      6
39        LEIA        MOTTI      1
40        LEIA       TARKIN      1
41         HAN         LEIA     13
42 DARTH VADER      OBI-WAN      1
43     DODONNA GOLD LEADER       1
44     DODONNA        WEDGE      1
45     DODONNA         LUKE      1
46 GOLD LEADER        WEDGE      1
47 GOLD LEADER         LUKE      1
48        LUKE        WEDGE      2
49       BIGGS         LEIA      1
50        LEIA  RED LEADER       1
51        LUKE  RED LEADER       3
52       BIGGS  RED LEADER       3
53       BIGGS        C-3PO      1
54       C-3PO  RED LEADER       1
55  RED LEADER        WEDGE      3
56 GOLD LEADER  RED LEADER       1
57       BIGGS        WEDGE      2
58  RED LEADER      RED TEN      1
59       BIGGS GOLD LEADER       1
60        LUKE      RED TEN      1
> head(data)
```

```
     source target weight
1     C-3PO  R2-D2     17
2      LUKE  R2-D2     13
3   OBI-WAN  R2-D2      6
4      LEIA  R2-D2      5
5       HAN  R2-D2      5
6 CHEWBACCA  R2-D2      3
```

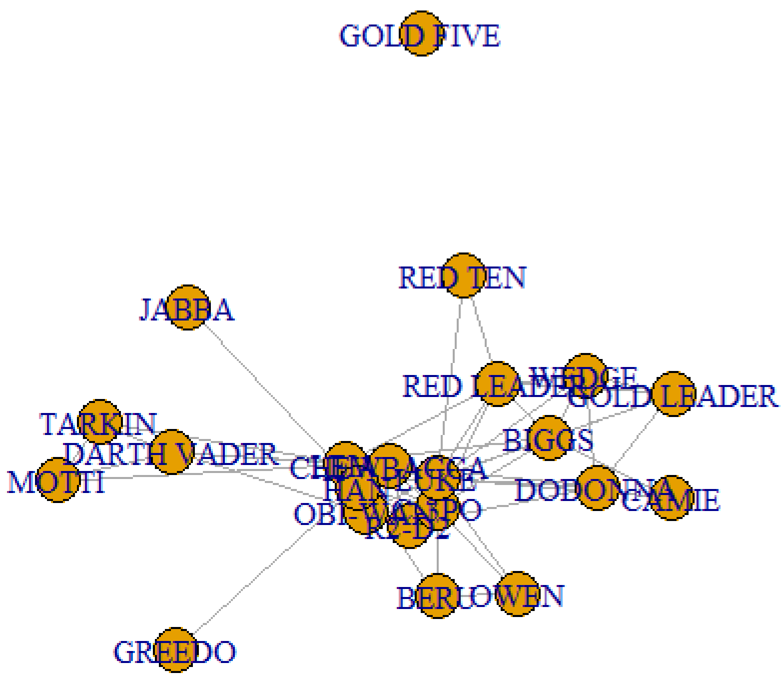
```
Data1 = Nodes
> data1 = read.csv(file.choose(),header=T)
> head(data1)
         name id
1       R2-D2  0
2   CHEWBACCA  1
3       C-3PO  2
4        LUKE  3
5 DARTH VADER  4
6       CAMIE  5

>

> library(igraph)
> g <- graph_from_data_frame(d=data, vertices=data1, directed=FALSE)
> g
IGRAPH d7d8fb2 UNW- 22 60 --
+ attr: name (v/c), id (v/n), weight (e/n)
+ edges from d7d8fb2 (vertex names):
 [1] R2-D2      --C-3PO       R2-D2      --LUKE         R2-D2      --OBI-WAN
 [4] R2-D2      --LEIA        R2-D2      --HAN          R2-D2      --CHEWBACCA
 [7] R2-D2      --DODONNA     CHEWBACCA  --OBI-WAN      CHEWBACCA  --C-3PO
[10] CHEWBACCA  --LUKE        CHEWBACCA  --HAN          CHEWBACCA  --LEIA
[13] CHEWBACCA  --DARTH VADER CHEWBACCA  --DODONNA      LUKE       --CAMIE
[16] CAMIE      --BIGGS       LUKE       --BIGGS        DARTH VADER--LEIA
[19] LUKE       --BERU        BERU       --OWEN         C-3PO      --BERU
[22] LUKE       --OWEN        C-3PO      --LUKE         C-3PO      --OWEN
+ ... omitted several edges

>

> V(g)
+ 22/22 vertices, named, from d7d8fb2:
 [1] R2-D2       CHEWBACCA   C-3PO       LUKE        DARTH VADER CAMIE       BIGGS
 [8] LEIA        BERU        OWEN        OBI-WAN     MOTTI       TARKIN      HAN
[15] GREEDO      JABBA       DODONNA     GOLD LEADER WEDGE       RED LEADER  RED TEN
[22] GOLD FIVE

>

> V(g)$name
 [1] "R2-D2"       "CHEWBACCA"   "C-3PO"       "LUKE"        "DARTH VADER" "CAMIE"
 [7] "BIGGS"       "LEIA"        "BERU"        "OWEN"        "OBI-WAN"     "MOTTI"
[13] "TARKIN"      "HAN"         "GREEDO"      "JABBA"       "DODONNA"     "GOLD LEADER"
[19] "WEDGE"       "RED LEADER"  "RED TEN"     "GOLD FIVE"

>

> vertex_attr(g)
$name
 [1] "R2-D2"       "CHEWBACCA"   "C-3PO"       "LUKE"        "DARTH VADER" "CAMIE"
 [7] "BIGGS"       "LEIA"        "BERU"        "OWEN"        "OBI-WAN"     "MOTTI"
[13] "TARKIN"      "HAN"         "GREEDO"      "JABBA"       "DODONNA"     "GOLD LEADER"
```

```
[19] "WEDGE"       "RED LEADER"  "RED TEN"     "GOLD FIVE"

$id
 [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21


>
> E(g)
+ 60/60 edges from d7d8fb2 (vertex names):
 [1] R2-D2       --C-3PO       R2-D2       --LUKE        R2-D2       --OBI-WAN
 [4] R2-D2       --LEIA        R2-D2       --HAN         R2-D2       --CHEWBACCA
 [7] R2-D2       --DODONNA     CHEWBACCA   --OBI-WAN     CHEWBACCA   --C-3PO
[10] CHEWBACCA   --LUKE        CHEWBACCA   --HAN         CHEWBACCA   --LEIA
[13] CHEWBACCA   --DARTH VADER CHEWBACCA   --DODONNA     LUKE        --CAMIE
[16] CAMIE       --BIGGS       LUKE        --BIGGS       DARTH VADER--LEIA
[19] LUKE        --BERU        BERU        --OWEN        C-3PO       --BERU
[22] LUKE        --OWEN        C-3PO       --LUKE        C-3PO       --OWEN
[25] C-3PO       --LEIA        LUKE        --LEIA        LEIA        --BERU
[28] LUKE        --OBI-WAN     C-3PO       --OBI-WAN     LEIA        --OBI-WAN
+ ... omitted several edges
> E(g)$weight
 [1] 17 13  6  5  5  3  1  7  5 16 19 11  1  1  2  2  4  1  3  3  2  3 18  2
6 17  1 19  6  1  2
[32]  1  7  9 26  1  1  6  1  1 13  1  1  1  1  1  1  2  1  1  3  3  1  1  3
1  2  1  1  1


> edge_attr(g)
$weight
 [1] 17 13  6  5  5  3  1  7  5 16 19 11  1  1  2  2  4  1  3  3  2  3 18  2
6 17  1 19  6  1  2
[32]  1  7  9 26  1  1  6  1  1 13  1  1  1  1  1  1  2  1  1  3  3  1  1  3
1  2  1  1  1


> g[]
22 x 22 sparse Matrix of class "dgCMatrix"
   [[ suppressing 22 column names ‘R2-D2’, ‘CHEWBACCA’, ‘C-3PO’ ... ]]

R2-D2        .  3 17 13 . . .  5 . .  6 . .  5 . . 1 . . . . .
CHEWBACCA    3  .  5 16 1 . . 11 . .  7 . . 19 . . 1 . . . . .
C-3PO       17  5  . 18 . . 1  6 2 2  6 . .  6 . . . . . 1 . .
LUKE        13 16 18  . . 2 4 17 3 3 19 . . 26 . . 1 1 2 3 1 .
DARTH VADER  .  1  .  . . . .  1 . .  1 1 7  . . . . . . . . .
CAMIE        .  .  .  2 . . 2  . . .  . . .  . . . . . . . . .
BIGGS        .  .  1  4 . 2 .  1 . .  . . .  . . . . 1 2 3 . .
LEIA         5 11  6 17 1 . 1  . 1 .  1 1 1 13 . . . . . 1 . .
BERU         .  .  2  3 . . .  1 . 3  . . .  . . . . . . . . .
OWEN         .  .  2  3 . . .  . 3 .  . . .  . . . . . . . . .
OBI-WAN      6  7  6 19 1 . .  1 . .  . . .  9 . . . . . . . .
MOTTI        .  .  .  . 1 . .  1 . .  . . 2  . . . . . . . . .
TARKIN       .  .  .  . 7 . .  1 . .  . 2 .  . . . . . . . . .
HAN          5 19  6 26 . . . 13 . .  9 . .  . 1 1 . . . . . .
GREEDO       .  .  .  . . . .  . . .  . . .  1 . . . . . . . .
JABBA        .  .  .  . . . .  . . .  . . .  1 . . . . . . . .
DODONNA      1  1  .  1 . . .  . . .  . . .  . . . . 1 1 . . .
```

```
GOLD LEADER   .  .  .  1 . . 1  . . .  . . .  . . . 1 . 1 1 . .
WEDGE         .  .  .  2 . . 2  . . .  . . .  . . . 1 1 . 3 . .
RED LEADER    .  .  1  3 . . 3  1 . .  . . .  . . . . 1 3 . 1 .
RED TEN       .  .  .  1 . . .  . . .  . . .  . . . . . . 1 . .
GOLD FIVE     .  .  .  . . . .  . . .  . . .  . . . . . . . . .
```

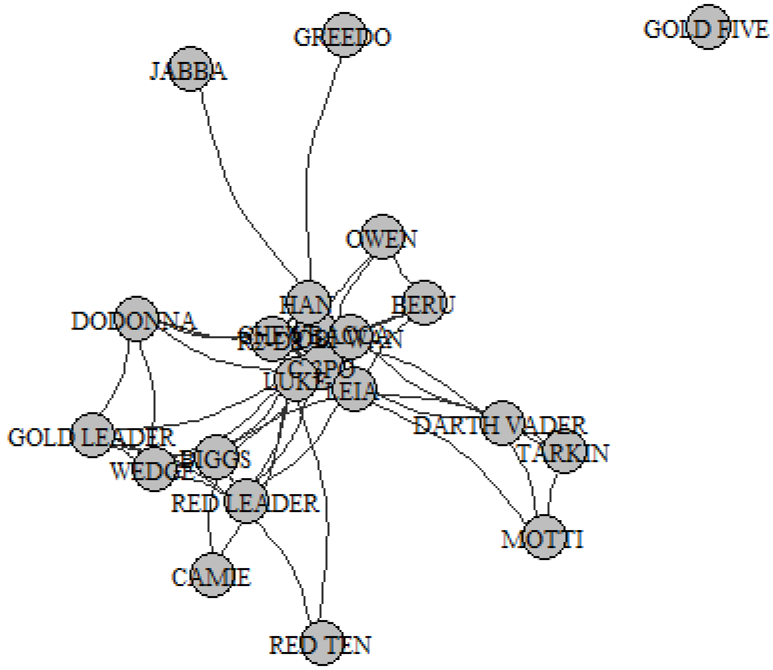
```
> g[1,]
      R2-D2   CHEWBACCA       C-3PO        LUKE DARTH VADER       CAMIE 
BIGGS        LEIA 
          0           3          17          13           0           0 
0           5 
       BERU        OWEN     OBI-WAN       MOTTI      TARKIN         HAN 
GREEDO       JABBA 
          0           0           6           0           0           5 
0          0 
    DODONNA GOLD LEADER       WEDGE  RED LEADER     RED TEN   GOLD FIVE 
          1           0           0           0           0           0 
```

```
par(mar=c(0,0,0,0))
> plot(g)
```

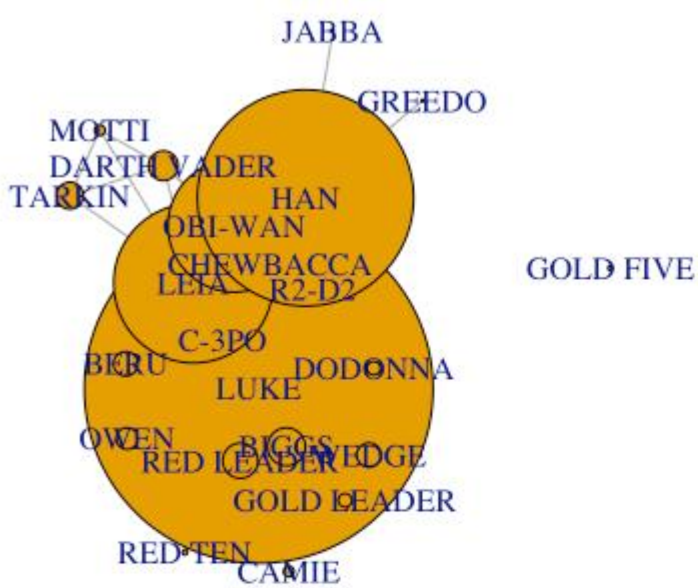



```
> par(mar=c(0,0,0,0))
> plot(g)
> par(mar=c(0,0,0,0))
> plot(g,
+      vertex.color = "grey", # change color of nodes
+      vertex.label.color = "black", # change color of labels
+      vertex.label.cex = .75, # change size of labels to 75% of original size
+      edge.curved=.25, # add a 25% curve to the edges
```

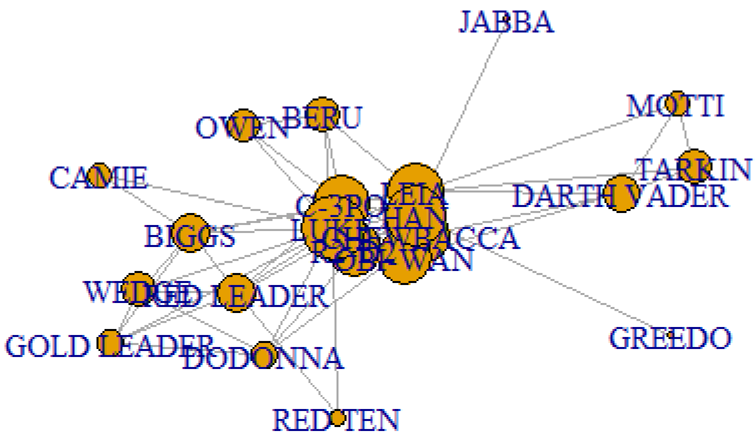
```
+      edge.color="grey20") # change edge color to grey
>
```



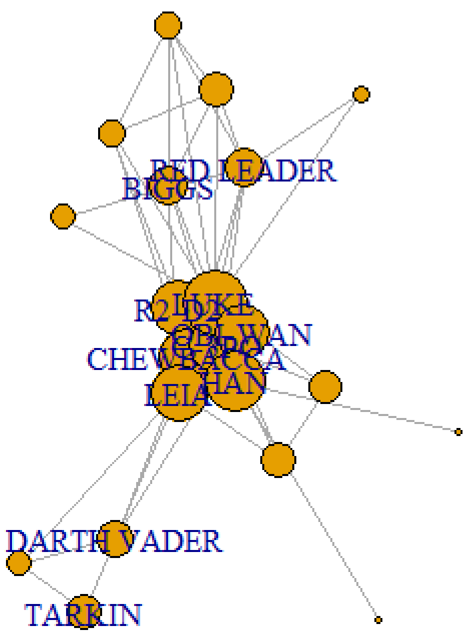

```
> V(g)$size <- strength(g)
> par(mar=c(0,0,0,0)); plot(g)
```



```
> V(g)$size <- log(strength(g)) * 4 + 3
> par(mar=c(0,0,0,0));plot(g)
```



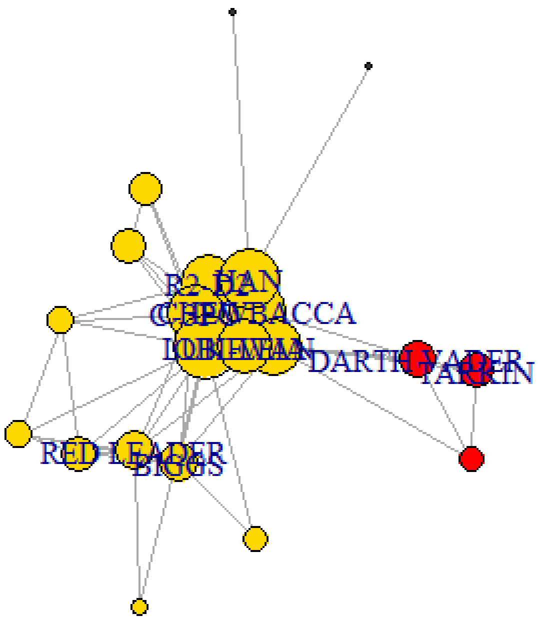

```
> V(g)$label <- ifelse( strength(g)>=10, V(g)$name, NA )
> par(mar=c(0,0,0,0)); plot(g)
```




```
> data1$name=="R2-D2"
 [1]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE
[16] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
> ifelse(data1$name=="R2-D2", "yes", "no")
```

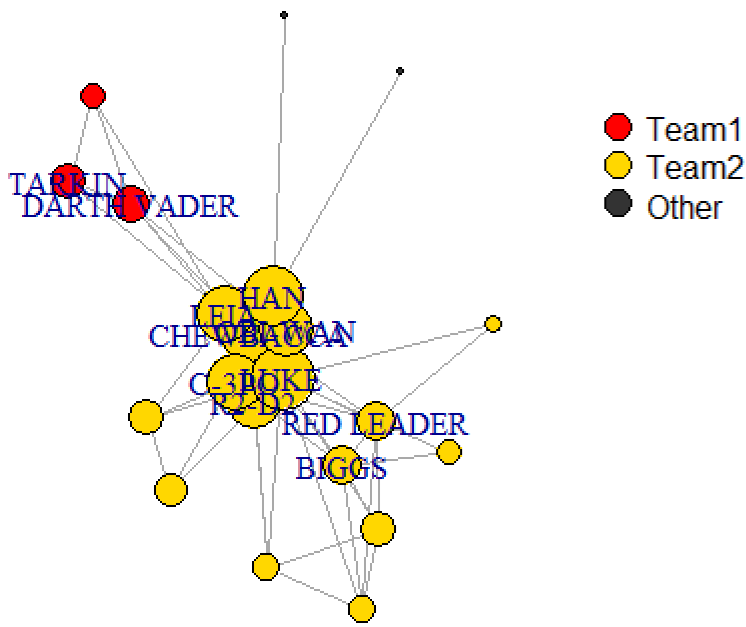
```
 [1] "yes" "no"  "no"  "no"  "no"  "no"  "no"  "no"  "no"  "no"  "no"  "no"
"no"  "no"  "no"
[16] "no"  "no"  "no"  "no"  "no"  "no"  "no"


> ifelse(grepl("R", data1$name), "yes", "no")
 [1] "yes" "no"  "no"  "no"  "yes" "no"  "no"  "no"  "yes" "no"  "no"  "no"
"yes" "no"  "yes"
[16] "no"  "no"  "yes" "no"  "yes" "yes" "no"


> Team1 <- c("DARTH VADER", "MOTTI", "TARKIN")
> Team2 <- c("R2-D2", "CHEWBACCA", "C-3PO", "LUKE", "CAMIE", "BIGGS","LEIA", "BERU",
"OWEN", "OBI-WAN", "HAN", "DODONNA","GOLD LEADER", "WEDGE", "RED LEADER", "RED TEN",
"GOLD FIVE")
> other <- c("GREEDO", "JABBA")
>
> V(g)$color <- NA
> V(g)$color[V(g)$name %in% Team1] <- "red"
> V(g)$color[V(g)$name %in% Team2] <- "gold"
> V(g)$color[V(g)$name %in% other] <- "grey20"
> vertex_attr(g)
$name
 [1] "R2-D2"         "CHEWBACCA"     "C-3PO"         "LUKE"          "DARTH VADER" "CAMIE"
 [7] "BIGGS"         "LEIA"          "BERU"          "OWEN"          "OBI-WAN"     "MOTTI"
[13] "TARKIN"        "HAN"           "GREEDO"        "JABBA"         "DODONNA"     "GOLD LEADER"
[19] "WEDGE"         "RED LEADER"    "RED TEN"       "GOLD FIVE"

$id
 [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21

$size
 [1] 18.648092 19.572539 19.635532 22.439250 12.591581  8.545177 13.556229 19.310150
11.788898
[10] 11.317766 18.567281  8.545177 12.210340 20.528107  3.000000  3.000000  9.437752
9.437752
[19] 11.788898 13.259797  5.772589      -Inf

$label
 [1] "R2-D2"         "CHEWBACCA"     "C-3PO"         "LUKE"          "DARTH VADER" NA
 [7] "BIGGS"         "LEIA"          NA              NA              "OBI-WAN"     NA
[13] "TARKIN"        "HAN"           NA              NA              NA            NA
[19] NA              "RED LEADER"    NA              NA

$color
 [1] "gold"   "gold"   "gold"   "gold"   "red"    "gold"   "gold"   "gold"   "gold"
"gold"
[11] "gold"   "red"    "red"    "gold"   "grey20" "grey20" "gold"   "gold"   "gold"
"gold"
[21] "gold"   "gold"


>

par(mar=c(0,0,0,0)); plot(g)
```



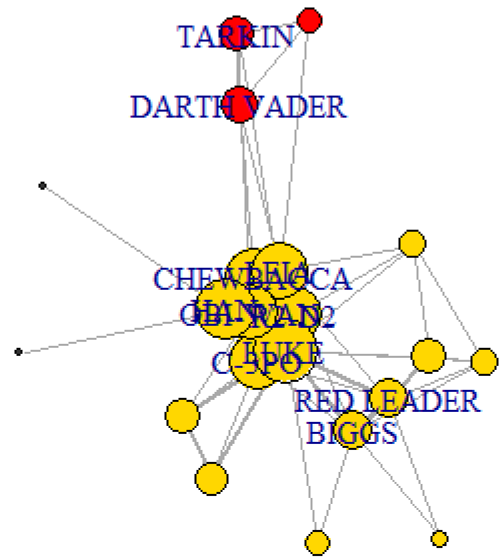
```
> 1 %in% c(1,2,3,4)
[1] TRUE
```

```
> 1 %in% c(2,3,4)
[1] FALSE
```

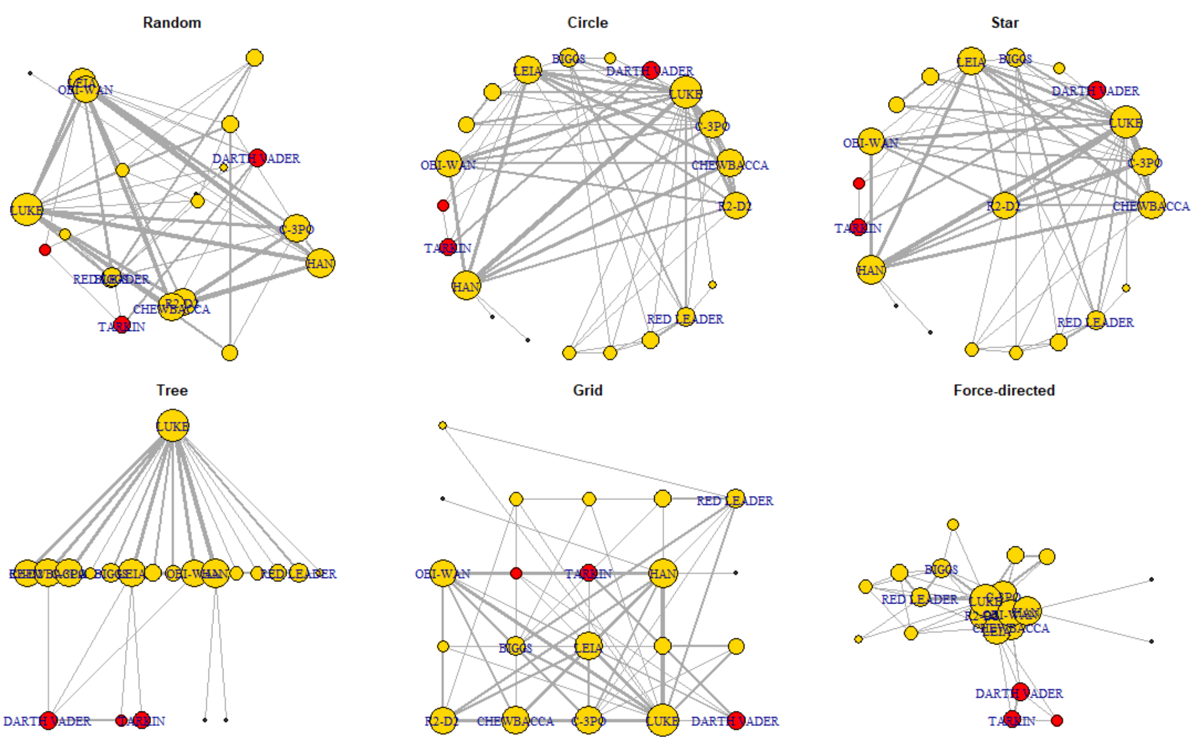
```
> par(mar=c(0,0,0,0)); plot(g)
> legend(x=.75, y=.75, legend=c("Team1", "Team2", "Other"), pch=21,
pt.bg=c("red", "gold", "grey20"), pt.cex=2, bty="n")
```




```
> E(g)$width <- log(E(g)$weight) + 1
> edge_attr(g)
$weight
 [1] 17 13  6  5  5  3  1  7  5 16 19 11  1  1  2  2  4  1  3  3  2  3 18  2
6 17  1 19  6  1  2
[32]  1  7  9 26  1  1  6  1  1 13  1  1  1  1  1  1  2  1  1  3  3  1  1  3
1  2  1  1  1

$width
 [1] 3.833213 3.564949 2.791759 2.609438 2.609438 2.098612 1.000000 2.945910
2.609438 3.772589
[11] 3.944439 3.397895 1.000000 1.000000 1.693147 1.693147 2.386294 1.000000
2.098612 2.098612
[21] 1.693147 2.098612 3.890372 1.693147 2.791759 3.833213 1.000000 3.944439
2.791759 1.000000
[31] 1.693147 1.000000 2.945910 3.197225 4.258097 1.000000 1.000000 2.791759
1.000000 1.000000
[41] 3.564949 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.693147
1.000000 1.000000
[51] 2.098612 2.098612 1.000000 1.000000 2.098612 1.000000 1.693147 1.000000
1.000000 1.000000


par(mar=c(0,0,0,0)); plot(g)
```



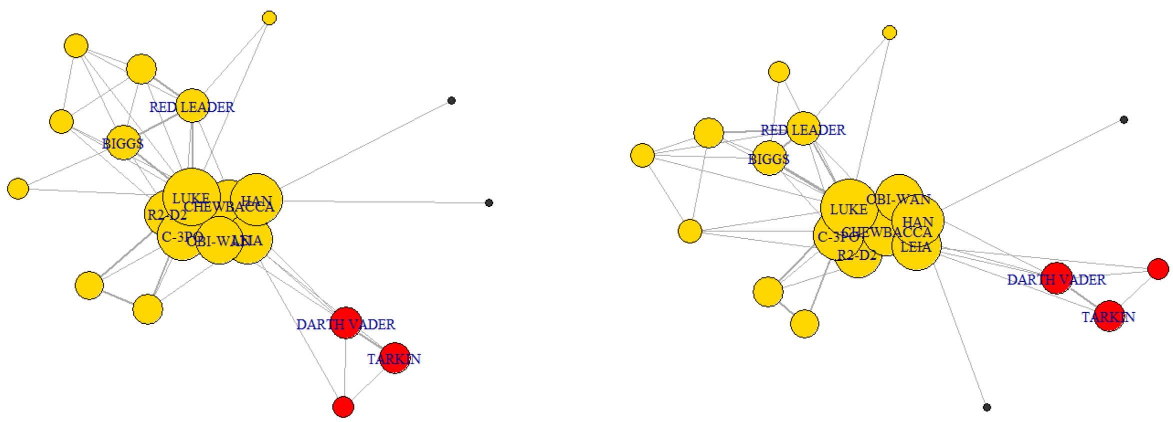

```
> par(mfrow=c(2, 3), mar=c(0,0,1,0))
> plot(g, layout=layout_randomly, main="Random")
> plot(g, layout=layout_in_circle, main="Circle")
> plot(g, layout=layout_as_star, main="Star")
> plot(g, layout=layout_as_tree, main="Tree")
> plot(g, layout=layout_on_grid, main="Grid")
> plot(g, layout=layout_with_fr, main="Force-directed")
```




```
> l <- layout_randomly(g)
```

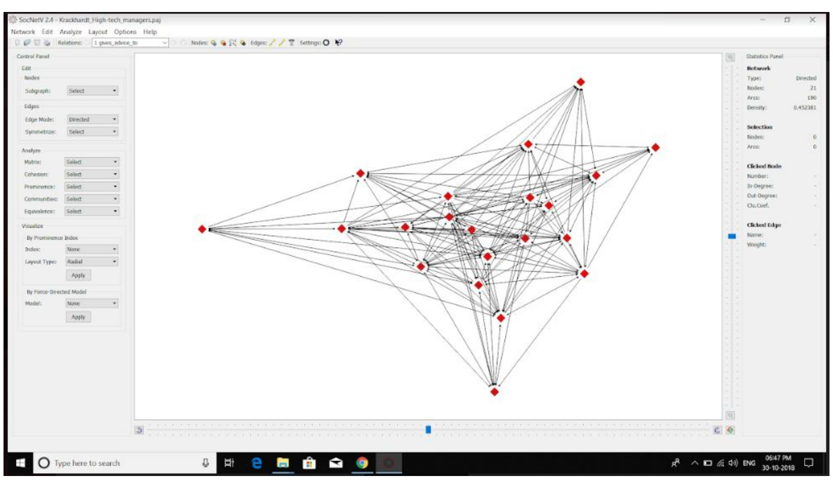
```
> str(l)
 num [1:22, 1:2] 0.809 0.429 -0.41 0.508 -0.469 ...
```

```
> par(mfrow=c(1,2))
> set.seed(777)
> fr <- layout_with_fr(g, niter=1000)
> par(mar=c(0,0,0,0)); plot(g, layout=fr)
> set.seed(666)
> fr <- layout_with_fr(g, niter=1000)
> par(mar=c(0,0,0,0)); plot(g, layout=fr)
```



b. SocNetV

SOCNETV



- Betweenness Centrality

BETWEENNESS CENTRALITY (BC)

Network name: Krackhardt's High-tech managers
Actors: 21

The BC index of a node u is the sum of $\delta_{(s,t,u)}$ for all s,t $\in V$
where $\delta_{(s,t,u)}$ is the ratio of all geodesics between s and t which run through u.
Read the Manual for more.
BC' is the standardized index (BC divided by (N-1)(N-2)/2 in symmetric nets or (N-1)(N-2) otherwise.

BC range: $0 \leq BC \leq 380$ **(Number of pairs of nodes excluding u)**

BC' range: $0 \leq BC' \leq 1$ **(BC'=1 when the node falls on all geodesics)**

Node	Label	BC	BC'	%BC'
1	v1	13.747	0.036	3.618
2	v2	5.936	0.016	1.562
3	v3	6.605	0.017	1.738
4	v4	13.709	0.036	3.608
5	v5	5.079	0.013	1.336
6	v6	0.000	0.000	0.000
7	v7	27.625	0.073	7.270
8	v8	3.975	0.010	1.046
9	v9	3.954	0.010	1.041
10	v10	18.297	0.048	4.815
11	v11	1.198	0.003	0.315
12	v12	0.254	0.001	0.067
13	v13	0.893	0.002	0.235
14	v14	0.589	0.002	0.155
15	v15	6.133	0.016	1.614
16	v16	0.700	0.002	0.184
17	v17	2.532	0.007	0.666
18	v18	88.917	0.234	23.399
19	v19	0.754	0.002	0.198
20	v20	7.979	0.021	2.100
21	v21	60.127	0.158	15.823

BC Sum = 269.000

Max BC' = 0.234 (node 18)
Min BC' = 0.000 (node 6)
BC' classes = 21

BC' Sum = 0.708
BC' Mean = 0.034
BC' Variance = 0.003

GROUP BETWEENNESS CENTRALIZATION (GBC)

GBC = 0.210

GBC range: $0 \leq GBC \leq 1$

GBC = 0, when all the nodes have exactly the same betweenness index.
GBC = 1, when one node falls on all other geodesics between all the remaining (N-1) nodes.
This is exactly the situation realised by a star graph.
(Wasserman & Faust, formula 5.13, p. 192)

Betweenness Centrality report,
Created by Social Network Visualizer v2.4: Tue, 30.Oct.2018 18:45:19
Computation time: 222 msecs

-

- Closeness Centrality

CLOSENESS CENTRALITY (CC) REPORT

Network name: Krackhardt's High-tech managers
Actors: 21

The CC index is the inverted sum of geodesic distances from each node u to all other nodes.
Note: The CC index considers outbound arcs only and isolate nodes are dropped by default.
Read the Manual for more.
CC' is the standardized index (CC multiplied by (N-1 minus isolates)).

CC range: $0 \leq CC \leq 0.05$ **(1 / Number of node pairs excluding u)**

CC' range: $0 \leq CC' \leq 1$ **(CC'=1 when a node is the center of a star graph)**

Node	Label	CC	CC'	%CC'
1	v1	0.029	0.588	58.824
2	v2	0.022	0.444	44.444
3	v3	0.040	0.800	80.000
4	v4	0.036	0.714	71.429
5	v5	0.040	0.800	80.000
6	v6	0.021	0.417	41.667
7	v7	0.031	0.625	62.500
8	v8	0.031	0.625	62.500
9	v9	0.037	0.741	74.074
10	v10	0.038	0.769	76.923
11	v11	0.022	0.444	44.444
12	v12	0.022	0.435	43.478
13	v13	0.029	0.588	58.824
14	v14	0.028	0.556	55.556
15	v15	0.050	1.000	100.000
16	v16	0.027	0.541	54.054
17	v17	0.025	0.500	50.000
18	v18	0.043	0.870	86.957
19	v19	0.034	0.690	68.966
20	v20	0.036	0.714	71.429
21	v21	0.034	0.690	68.966

CC Sum = 0.678

Max CC' = 1.000 (node 15)
Min CC' = 0.417 (node 6)
CC' classes = 15

CC' Sum = 13.550
CC' Mean = 0.645
CC' Variance = 0.023

GROUP CLOSENESS CENTRALIZATION (GCC)

GCC = 0.765

GCC range: $0 \leq GCC \leq 1$

GCC = 0, when the lengths of the geodesics are all equal, i.e. a complete or a circle graph.
GCC = 1, when one node has geodesics of length 1 to all the other nodes, and the other nodes have geodesics of length 2. to the remaining (N-2) nodes.
This is exactly the situation realised by a star graph.
(Wasserman & Faust, formula 5.9, p. 186-187)

Closeness Centrality report,
Created by [Social Network Visualizer] v2.4: Tue, 30.Oct.2018 18:45:01
Computation time: 311 msecs

- Degree Centrality

30/10/2018 socnetv-report-centrality-out-degree-18-10-30-184454.html

DEGREE CENTRALITY (DC) REPORT

Network name: Krackhardt's High-tech managers
Actors: 21

In undirected networks, the DC index is the sum of edges attached to a node u.
In directed networks, the index is the sum of outbound arcs from node u to all adjacent nodes (also called "outDegree Centrality").
If the network is weighted, the DC score is the sum of weights of outbound edges from node u to all adjacent nodes.
Note: To compute inDegree Centrality, use the Degree Prestige measure.
DC' is the standardized index (DC divided by N-1 (non-valued nets) or by sumDC (valued nets).

DC range: $0 \leq DC \leq 20$

DC' range: $0 \leq DC' \leq 1$

Node↕	Label↕	DC↕	DC'↕	%DC'↕
1	v1	6.000	0.300	30.000
2	v2	3.000	0.150	15.000
3	v3	15.000	0.750	75.000
4	v4	12.000	0.600	60.000
5	v5	15.000	0.750	75.000
6	v6	1.000	0.050	5.000
7	v7	8.000	0.400	40.000
8	v8	8.000	0.400	40.000
9	v9	13.000	0.650	65.000
10	v10	14.000	0.700	70.000
11	v11	3.000	0.150	15.000
12	v12	2.000	0.100	10.000
13	v13	6.000	0.300	30.000
14	v14	4.000	0.200	20.000
15	v15	20.000	1.000	100.000
16	v16	4.000	0.200	20.000
17	v17	5.000	0.250	25.000
18	v18	17.000	0.850	85.000
19	v19	11.000	0.550	55.000
20	v20	12.000	0.600	60.000
21	v21	11.000	0.550	55.000

DC Sum = 190.000

Max DC' = 1.000 (node 15)
Min DC' = 0.050 (node 6)
DC' classes = 14

DC' Sum = 9.500
DC' Mean = 0.452
DC' Variance = 0.071

GROUP DEGREE CENTRALIZATION (GDC)

GDC = 0.575

GDC range: $0 \leq GDC \leq 1$

GDC = 0, when all out-degrees are equal (i.e. regular lattice).
GDC = 1, when one node completely dominates or overshadows the other nodes.
(Wasserman & Faust, formula 5.5, p. 177)
(Wasserman & Faust, p. 101)

Degree Centrality report,
Created by Social Network Visualizer v2.4: Tue, 30.Oct.2018 18:44:54
Computation time: 221 msecs

-

- Histogram

HIERARCHICAL CLUSTERING (HCA)

Network name: Krackhardt's High-tech managers
Actors: 21

Input matrix: Distances

Distance/dissimilarity metric: Euclidean distance

Clustering method/criterion: Average-linkage (UPGMA)

Analysis results

Structural Equivalence Matrix:

Actor/Actor	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.000	6.403	5.916	4.690	5.745	7.483	6.403	4.899	5.657	5.196	5.568	7.483	5.568	5.477	6.083	3.742	5.385	4.899	6.000	5.385	6.164
2	6.403	0.000	7.071	6.557	8.246	4.583	4.899	5.568	7.937	8.246	5.099	5.568	7.874	5.196	8.832	7.000	5.099	7.141	8.185	6.782	5.196
3	5.916	7.071	0.000	4.359	5.292	7.141	5.292	4.796	4.583	4.899	7.211	6.708	5.831	5.196	4.472	6.403	5.831	4.796	5.000	4.243	4.796
4	4.690	6.557	4.359	0.000	5.916	6.782	5.568	3.742	5.657	4.796	6.403	6.633	6.403	6.000	5.568	5.477	5.000	4.899	6.325	4.359	4.472
5	5.745	8.246	5.292	5.916	0.000	8.426	6.481	5.568	4.796	4.243	7.483	8.426	4.000	5.568	4.000	5.385	7.211	5.000	3.606	4.899	6.856
6	7.483	4.583	7.141	6.782	8.426	0.000	5.916	5.831	7.874	8.660	6.403	4.690	7.937	5.831	8.888	7.483	5.385	8.124	8.485	6.708	5.292
7	6.403	4.899	5.292	5.568	6.481	5.916	0.000	5.000	6.083	6.928	5.292	5.385	6.928	3.873	6.782	6.856	5.099	5.385	6.557	5.657	3.873
8	4.899	5.568	4.796	3.742	5.568	5.831	5.000	0.000	5.292	5.000	5.568	6.164	6.083	5.099	5.745	5.292	5.385	4.472	6.000	4.796	4.243
9	5.657	7.937	4.583	5.657	4.796	7.874	6.083	5.292	0.000	5.385	7.000	7.348	4.796	5.292	4.583	5.477	6.708	5.477	5.477	5.385	6.481
10	5.196	8.246	4.899	4.796	4.243	8.660	6.928	5.000	5.385	0.000	7.483	8.660	5.292	6.403	4.243	4.583	6.782	4.359	4.123	5.099	6.856
11	5.568	5.099	7.211	6.403	7.483	6.403	5.292	5.568	7.000	7.483	0.000	5.745	7.211	5.745	8.124	6.083	5.099	7.280	7.280	6.633	7.000
12	7.483	5.568	6.708	6.633	8.426	4.690	5.385	6.164	7.348	8.660	5.745	0.000	7.810	5.657	8.426	7.874	5.196	8.124	8.124	6.557	5.477
13	5.568	7.874	5.831	6.403	4.000	7.937	6.928	6.083	4.796	5.292	7.211	7.810	0.000	5.196	5.099	5.196	7.211	5.568	4.796	5.831	7.280
14	5.477	5.196	5.196	6.000	5.568	5.831	3.873	5.099	5.292	6.403	5.745	5.657	5.196	0.000	6.083	5.657	5.385	4.690	5.477	5.000	4.472
15	6.083	8.832	4.472	5.568	4.000	8.888	6.782	5.745	4.583	4.243	8.124	8.426	5.099	6.083	0.000	5.916	7.616	4.796	3.873	4.472	6.708
16	3.742	7.000	6.403	5.477	5.385	7.483	6.856	5.292	5.477	4.583	6.083	7.874	5.196	5.657	5.916	0.000	5.916	5.292	5.477	5.568	6.928
17	5.385	5.099	5.831	5.000	7.211	5.385	5.099	5.385	6.708	6.782	5.099	5.196	7.211	5.385	7.616	5.916	0.000	7.000	7.416	5.657	5.000
18	4.899	7.141	4.796	4.899	5.000	8.124	5.385	4.472	5.477	4.359	7.280	8.124	5.568	4.690	4.796	5.292	7.000	0.000	5.099	4.796	4.899
19	6.000	8.185	5.000	6.325	3.606	8.485	6.557	6.000	5.477	4.123	7.280	8.124	4.796	5.477	3.873	5.477	7.416	5.099	0.000	5.196	7.211
20	5.385	6.782	4.243	4.359	4.899	6.708	5.657	4.796	5.385	5.099	6.633	6.557	5.831	5.000	4.472	5.568	5.657	4.796	5.196	0.000	4.796
21	6.164	5.196	4.796	4.472	6.856	5.292	3.873	4.243	6.481	6.856	7.000	5.477	7.280	4.472	6.708	6.928	5.000	4.899	7.211	4.796	0.000

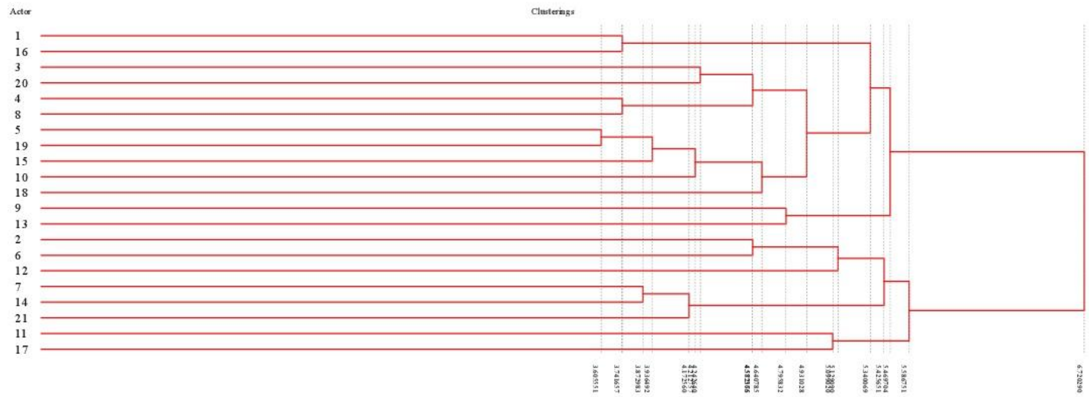
Values: real numbers (printed decimals 3)
- **Max value:** 8.88819
- **Min value:** 0

Hierarchical Clustering of Equivalence Matrix:

```
Seq     Level   Actors
1       3.606   5 19
2       3.742   1 16
3       3.742   4 8
4       3.873   7 14
5       3.936   5 19 15
6       4.173   7 14 21
7       4.213   5 19 15 10
8       4.243   3 20
9       4.577   3 20 4 8
10      4.583   2 6
11      4.641   5 19 15 10 18
12      4.796   9 13
13      4.931   3 20 4 8 5 19 15 10 18
14      5.099   11 17
15      5.129   2 6 12
16      5.340   1 16 3 20 4 8 5 19 15 10 18
17      5.426   2 6 12 7 14 21
18      5.470   1 16 3 20 4 8 5 19 15 10 18 9 13
19      5.587   2 6 12 7 14 21 11 17
20      6.720   1 16 3 20 4 8 5 19 15 10 18 9 13 2 6 12 7 14 21 11 17
```

Clustering Dendrogram (SVG)

30/10/2018 socnetv-report-equivalence-hierarchical-clustering-18-10-30-183751.html



Hierarchical Cluster Analysis report,
Created by [Social Network Visualizer] v2.4: Tue, 30.Oct.2018 18:37:51
Computation time: 371 msecs

PAJEK

```
*Network Hi-tech.net
*Vertices     36
      1 "Abe"                                         0.9606    0.5602
0.5000
      2 "Bob"                                         0.2207    0.5480
0.5000
      3 "Carl"                                        0.8044    0.3758
0.5000
      4 "Dale"                                        0.4005    0.5509
0.5000
      5 "Ev"                                          0.5539    0.7364
0.5000
      6 "Fred"                                        0.7905    0.4945
0.5000
      7 "Gary"                                        0.2694    0.5015
0.5000
      8 "Hal"                                         0.9307    0.3724
0.5000
      9 "Ivo"                                         0.4473    0.2931
0.5000
     10 "Jack"                                        0.0926    0.6923
0.5000
     11 "Ken"                                         0.3902    0.4205
0.5000
     12 "Len"                                         0.6873    0.3253
0.5000
     13 "Mel"                                         0.4204    0.6336
0.5000
     14 "Nan"                                         0.2949    0.3739
0.5000
     15 "Ovid"                                        0.1949    0.3849
0.5000
```

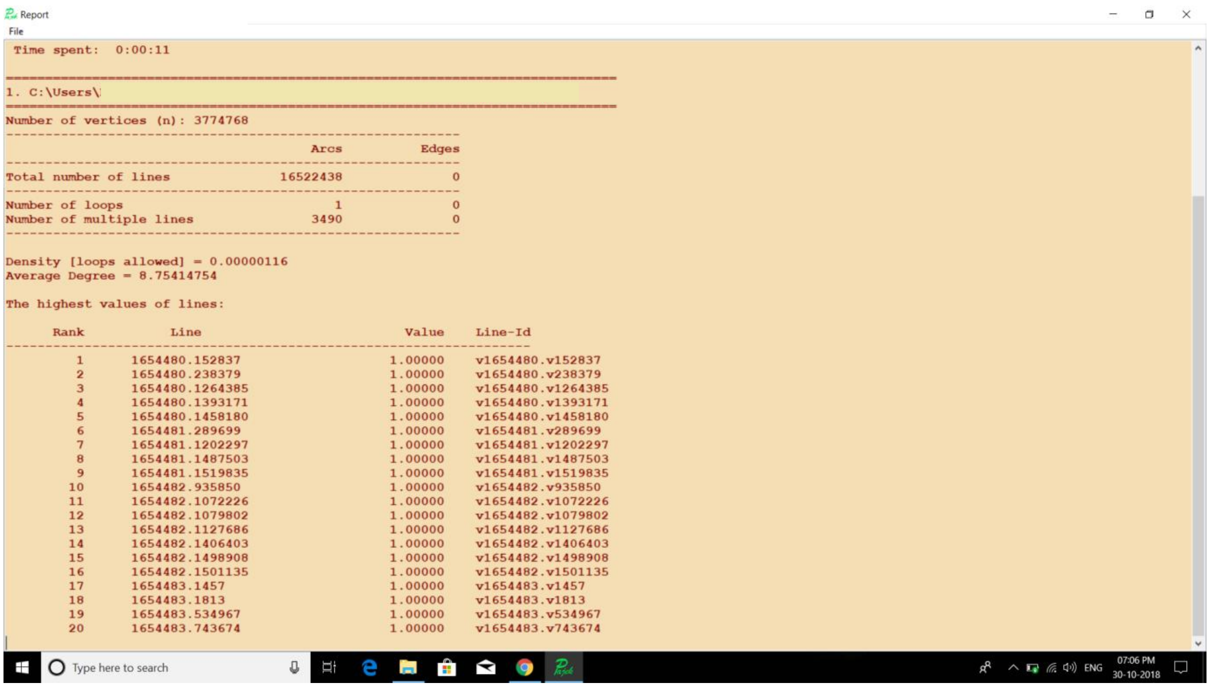
```
     16 "Pat"                                          0.6282    0.6977
0.5000
     17 "Quincy"                                       0.1781    0.8435
0.5000
     18 "Robin"                                        0.2682    0.6103
0.5000
     19 "Steve"                                        0.6568    0.5843
0.5000
     20 "Tom"                                          0.4703    0.3929
0.5000
     21 "Upton"                                        0.5872    0.4354
0.5000
     22 "Vic"                                          0.3767    0.2147
0.5000
     23 "Walt"                                         0.3751    0.7260
0.5000
     24 "Rick"                                         0.4737    0.5500
0.5000
     25 "York"                                         0.2920    0.8391
0.5000
     26 "Zoe"                                          0.6107    0.2845
0.5000
     27 "Alex"                                         0.3398    0.6788
0.5000
     28 "Ben"                                          0.0394    0.5201
0.5000
     29 "Chris"                                        0.3718    0.4928
0.5000
     30 "Dan"                                          0.5529    0.5847
0.5000
     31 "Earl"                                         0.7588    0.7272
0.5000
     32 "Fran"                                         0.4003    0.8406
0.5000
     33 "Gerry"                                        0.5240    0.4304
0.5000
     34 "Hugh"                                         0.3706    0.3380
0.5000
     35 "Irv"                                          0.6422    0.4854
0.5000
     36 "Jim"                                          0.6702    0.8893
0.5000
*Arcs
     10      2        1
     28      2        1
      2     10        1
      2      4        1
      2     29        1
      2     15        1
     23     24        1
     23     29        1
     15     29        1
     15     14        1
     15     34        1
```

7	4	1
7	24	1
14	2	1
14	7	1
14	29	1
14	11	1
14	9	1
14	15	1
34	15	1
34	14	1
34	29	1
34	24	1
34	11	1
34	33	1
34	20	1
29	23	1
29	7	1
29	2	1
29	18	1
29	27	1
29	4	1
29	13	1
29	24	1
29	11	1
29	20	1
29	9	1
29	34	1
29	14	1
29	15	1
18	27	1
18	13	1
18	11	1
18	29	1
27	18	1
27	4	1
27	24	1
4	2	1
4	27	1
4	13	1
4	35	1
4	24	1
4	20	1
4	29	1
13	18	1
13	16	1
13	30	1
13	20	1
13	29	1
13	4	1
13	2	1
24	4	1
24	30	1
24	5	1
24	19	1

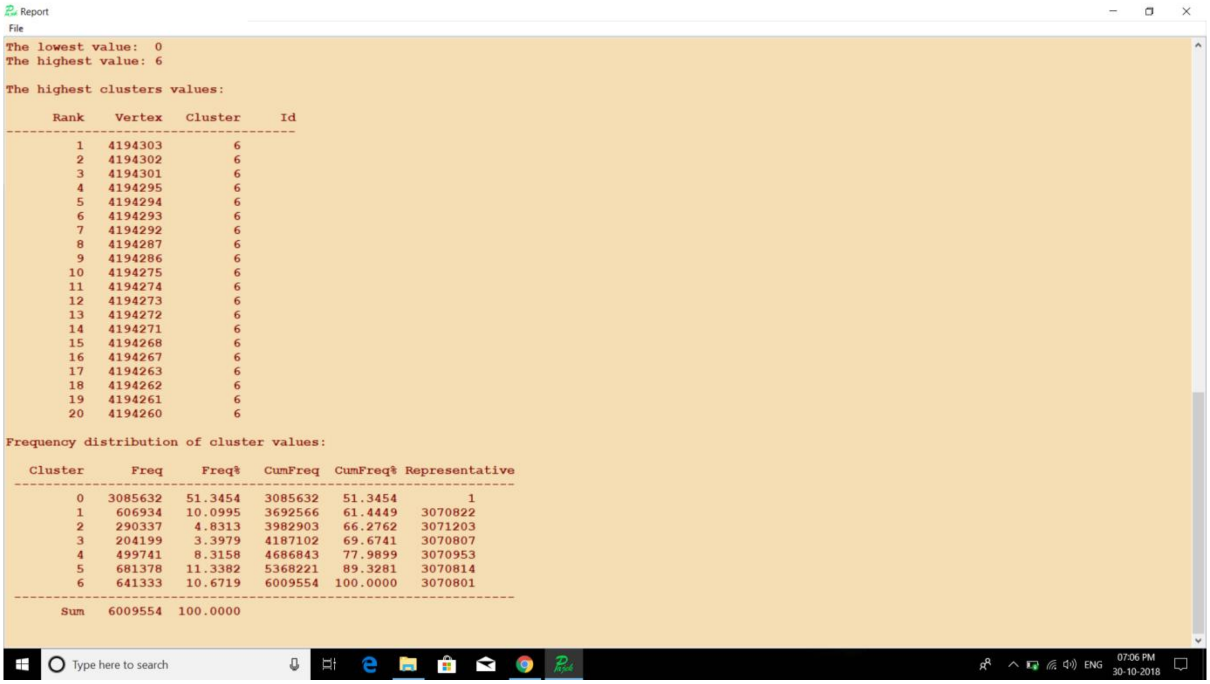
24	21	1
24	20	1
24	11	1
24	29	1
24	7	1
11	18	1
11	24	1
11	30	1
11	33	1
11	20	1
11	34	1
11	14	1
20	29	1
20	11	1
20	4	1
20	24	1
20	13	1
20	33	1
20	21	1
20	26	1
20	22	1
20	34	1
22	34	1
22	11	1
22	20	1
9	29	1
9	20	1
21	9	1
21	20	1
29	21	1
21	19	1
21	6	1
33	24	1
33	35	1
33	20	1
33	34	1
33	14	1
33	11	1
35	33	1
35	4	1
35	30	1
35	16	1
35	19	1
35	12	1
35	26	1
30	13	1
30	19	1
30	35	1
30	11	1
30	24	1
16	36	1
16	19	1
16	35	1
16	13	1

```
     36     16       1
     31     16       1
     31     19       1
      5     19       1
     19     30       1
     19     16       1
     19      5       1
     19     35       1
     19     33       1
     19     24       1
     12     33       1
     12     35       1
     12      3       1
     12     26       1
     26     21       1
     26     35       1
      6     21       1
      6     19       1
      1      6       1
      8      3       1
      8      6       1
      3      8       1
      3      6       1
      3     12       1
      3     35       1
     33     29       1
     29     33       1
     14     33       1
*Edges

*Partition Hi-tech_union.clu
*Vertices 36
      0
      0
      0
      2
      0
      0
      0
      1
      1
      1
      0
      0
      2
      0
      1
      3
      0
      2
      3
      0
      0
      0
```

0
0
0
0
0
0
1
0
0
0
0
0
0
3



```
Report
File
 Time spent:  0:00:11

==============================================================================
1. C:\Users\
==============================================================================
Number of vertices (n): 3774768
----------------------------------------------------------
                                        Arcs         Edges
----------------------------------------------------------
Total number of lines               16522438             0
----------------------------------------------------------
Number of loops                            1             0
Number of multiple lines                3490             0
----------------------------------------------------------

Density [loops allowed] = 0.00000116
Average Degree = 8.75414754

The highest values of lines:

      Rank           Line                          Value    Line-Id
-------------------------------------------------------------------
         1      1654480.152837                   1.00000    v1654480.v152837
         2      1654480.238379                   1.00000    v1654480.v238379
         3      1654480.1264385                  1.00000    v1654480.v1264385
         4      1654480.1393171                  1.00000    v1654480.v1393171
         5      1654480.1458180                  1.00000    v1654480.v1458180
         6      1654481.289699                   1.00000    v1654481.v289699
         7      1654481.1202297                  1.00000    v1654481.v1202297
         8      1654481.1487503                  1.00000    v1654481.v1487503
         9      1654481.1519835                  1.00000    v1654481.v1519835
        10      1654482.935850                   1.00000    v1654482.v935850
        11      1654482.1072226                  1.00000    v1654482.v1072226
        12      1654482.1079802                  1.00000    v1654482.v1079802
        13      1654482.1127686                  1.00000    v1654482.v1127686
        14      1654482.1406403                  1.00000    v1654482.v1406403
        15      1654482.1498908                  1.00000    v1654482.v1498908
        16      1654482.1501135                  1.00000    v1654482.v1501135
        17      1654483.1457                     1.00000    v1654483.v1457
        18      1654483.1813                     1.00000    v1654483.v1813
        19      1654483.534967                   1.00000    v1654483.v534967
        20      1654483.743674                   1.00000    v1654483.v743674
```


Report

File

The lowest value: 0
The highest value: 6

The highest clusters values:

Rank	Vertex	Cluster	Id
1	4194303	6	
2	4194302	6	
3	4194301	6	
4	4194295	6	
5	4194294	6	
6	4194293	6	
7	4194292	6	
8	4194287	6	
9	4194286	6	
10	4194275	6	
11	4194274	6	
12	4194273	6	
13	4194272	6	
14	4194271	6	
15	4194268	6	
16	4194267	6	
17	4194263	6	
18	4194262	6	
19	4194261	6	
20	4194260	6	

Frequency distribution of cluster values:

Cluster	Freq	Freq%	CumFreq	CumFreq%	Representative
0	3085632	51.3454	3085632	51.3454	1
1	606934	10.0995	3692566	61.4449	3070822
2	290337	4.8313	3982903	66.2762	3071203
3	204199	3.3979	4187102	69.6741	3070807
4	499741	8.3158	4686843	77.9899	3070953
5	681378	11.3382	5368221	89.3281	3070814
6	641333	10.6719	6009554	100.0000	3070801
Sum	6009554	100.0000			

2.

a.

In the paper, we present an arrangement of trials and investigations on anticipating the gender orientation of Twitter clients dependent on dialect free highlights separated either from the content or the metadata of clients' tweets. We play out our examinations on the Twisty dataset containing manual sex comments for clients talking six unique dialects. Our order results demonstrate that, while the forecast display dependent on dialect autonomous highlights performs more awful than the pack of-words demonstrate when preparing and testing on a similar dialect, it frequently beats the sack of-words show when connected to various dialects, indicating extremely stable outcomes crosswise over different dialects. At long last, we play out a similar investigation of highlight impact sizes over the six dialects and demonstrate that distinctions in our highlights compare to social separations.

Sexual orientation expectation is an entrenched undertaking in a creator profiling, valuable for a progression of downstream examinations and additionally prescient model enhancements. Most existing work on anticipating sexual orientation centers on misusing the phonetic generation of the clients, just seldom utilizing nonlinguistic data, for example, metadata or visual data. In this paper, we examine the likelihood of foreseeing sexual orientation of a Twitter client paying little heed to the dialect utilized in his or her tweets. We play out our examinations on a current dataset of Twitter clients talking six distinct dialects that were physically clarified for their sex. Our dialect free sexual orientation indicator depends on general semantic highlights, for example, the utilization of accentuation, and non-etymological highlights ascertained from Twitter metadata, for example, the client communication through answering, retweeting and favorite, time of posting, shading decisions, customer use and so on. The capability of a dialect-free technique for sexual orientation expectation is considered both for the field of normal dialect handling where utilizing additional phonetic factors is as of now picking up energy, and orders from sociologies and the humanities working with user-generated content, where such factors have a long convention. We trust that building such language-independent systems is the main tractable method for pushing ahead given the number of various dialects utilized in online life and the presence of preparing information just for a couple of high-thickness dialects. In the following area we quickly portray the dataset we played out our examinations on, in Section 3 we depict our dialect autonomous highlights, in Section 4 we give the test setup of our sex expectation tests, while in Section 5 we present the sex forecast results, and in addition a progression of investigations of the element spaces crosswise over dialects. In Section 6 we give a few ends and bearings for further research.

In this paper, we have exhibited a first keep running at the issue of dialect free sexual orientation distinguishing proof among Twitter clients. We have demonstrated that with dialect free highlights in the cross-lingual setting we consistently beat the sack of-words standard, and, besides, that the dialect autonomous models have a ten times littler F1 change, which turns out to be more hearty than the pack of-words models, and in this manner all the more dependably pertinent to new dialects. We have dissected the impact sizes of particular highlights among dialects and have demonstrated that our highlights consistently relate crosswise over dialects which likewise clarifies why the models work dependably crosswise over dialects. By performing various leveled bunching over dialects spoke to through element impact sizes we have demonstrated that the distinction in highlight esteems crosswise over dialects compares to the social separations of the speakers of those dialects. While the outcomes introduced in this paper are promising, there is a progression of open inquiries that must be investigated. The most squeezing one is the representativeness of clients in the Twisty corpus as they are Twitter clients that have self-detailed their identity test results. A method for estimating this representativeness is to apply these models to another sexual orientation expectation dataset. Additionally, highlights ought to likewise be investigated (arrange based, picture content and soon.), and in addition the capability of building extra dialect free creator profiling models, for example, age or instructive level indicators.

b.

Twitter has pulled in a huge number of clients that produce a humongous stream of data at a consistent pace. The examination network has in this manner begun proposing devices to remove important data from tweets. In this paper, we take an alternate point from the standard of past works: we unequivocally focus on the investigation of the course of events of tweets from "single clients". We characterize a system - named TUCAN - to look at data advertised by the objective clients after some time and to pinpoint repetitive subjects or subjects of intrigue. To begin with, tweets having a place with a similar time window are accumulated into "fledgling melodies". A few sifting techniques can be chosen to expel stop-words and lessen commotion. At that point, each combine of fledgling melodies is contrasted utilizing a closeness score with naturally feature the most widely recognized terms, in this way featuring repetitive or relentless subjects. TUCAN can be normally connected to look at fledgling melody sets produced from courses of events of various clients. By demonstrating real outcomes for both open profiles and mysterious clients, we demonstrate how TUCAN is helpful to feature significant data from an objective client's Twitter timetable. I. Presentation AND MOTIVATION Twitter is these days part of everybody's life, with hundreds of a huge number of individuals utilizing it on standard premise. Initially conceived as a microblogging administration, Twitter is currently being utilized to talk, to examine, to run surveys, to gather input, and so forth. It is not astonishing then that the enthusiasm of the examination network has been pulled in to contemplate the "social viewpoints" of Twitter. The client also uses portrayal point examination and network level social intrigue ID have as of late risen as hot research points. A large portion of past works centers on the investigation of "a network of twitters", whose tweets are dissected utilizing content and information mining methods to distinguish the points, temperaments, or interests. In this paper we take an alternate point: first, we center on the investigation of a Twitter target client. We think about the arrangement of tweets that show up on his Twitter open page, i.e., the objective client's course of events, and characterize a philosophy to investigate uncovered substance furthermore, separate conceivable important data. Which are the tweets that convey the most profitable data? Which are the themes he/she is intrigued into? How do this themes change after some time? Our second objective is to analyze the Twitter action of two (or more) target clients. Do they share some basic attributes? Is there any common intrigue? How imperative is for one client a subject of enthusiasm for the other client? What is the most widely recognized intrigue of these two clients, paying little respect to the time they are intrigued in it? We propose a graphical system which we term as **TUCAN - Twitter User Centric ANalyzer**. TUCAN features connections among tweets utilizing natural perception, permitting investigation of the data uncovered in them, hence empowering the extraction of profitable data from the client's timetable. From a philosophy stance, we expand upon content mining systems, adjusting them to adapt to the particular Twitter qualities. As info, we aggregate the objective client's tweets dependent on a window of time (e.g., multi-day, or seven days) so to shape fowl tunes, one for each time window. At the subsequent stage, sifting is connected to each feathered creature tune utilizing basic stop-word expulsion, stemming, lemmatization, or more confused changes in light of lexical databases. Next, terms in winged animal tunes are scored utilizing exemplary Term Frequency-Inverse Document Frequency (TF-IDF) to pinpoint those terms that are especially essential for the objective client. Each combine of winged creatures tunes are at last thought about by registering a similitude score, so to uncover those winged animal tunes that contain covering, and in this manner tireless, subjects. The yield is then spoken to utilizing a hued network, in which cell shading speaks to the similitude score. Thus, TUCAN offers a basic and regular visual portrayal of separated data that effectively uncovers the most fascinating flying creature tunes and the persevering themes the objective client is intrigued into amid a given day and age. In addition, correlations among flying creature tunes give instincts on the change of client interests and also the hugeness of points to the client. The system is normally stretched out to discover and separate likenesses among tweets of at least two target clients. TUCAN registers and graphically demonstrates the closeness among winged creature melodies produced from the courses of events of the sets of target clients, uncovering likenesses and basic interests that are available conceivably amid various eras. In this paper, we displayed TUCAN, a system to graphically speak to semantic connections of individual Twitter clients' timetables. Expanding on content mining procedures, TUCAN investigations "winged creature tunes", i.e., gathering of tweets having a place with the same day and age, and thinks about their similitude. The investigator is offered a GUI to research the effect of various preprocessing also, likeness definitions. Examinations directed on real Twitter clients demonstrate the capacity to pinpoint repetitive subjects, or relationships among clients.

c.

Group investigation is a test of information examination that concentrates fundamental examples in information. One application of bunch examination is in content mining, the investigation of vast accumulations of content to and likenesses between reports. We utilized a gathering of around 30,000 tweets separated from Twitter just before the World Container began. A typical issue with certifiable content information is the nearness of semantic commotion. In our case it would be unessential tweets that are inconsequential to predominant subjects. To battle this issue, we made a calculation that consolidated the DBSCAN calculation and an accord lattice. Along these lines we are left with the tweets that are identified with those prevailing topics. We at that point utilized group investigation to and those subjects that the tweets depict. We grouped the tweets utilizing k-implies, a generally utilized grouping calculation, and Non-Negative Matrix Factorization (NMF) and thought about the outcomes. The two calculations gave comparative outcomes, yet NMF ended up being quicker and given all the more effectively deciphered results. We investigated our outcomes utilizing two representation instruments, Gephi and Wordle. Keywords like k-implies, Non-Negative Matrix Factorization, group investigation, tex etc. We utilized bunch examination to and themes in the accumulation of tweets. NMF ended up being quicker and given more effortlessly deciphered outcomes. NMF chose a solitary tweet that spoke to a whole subject while k-implies can just give the tweets in every theme. Encourage perception methods are fundamental for deciphering the implications of the groups given by k-implies. There is still more to investigate with understanding content information in this way. We just took a gander at NMF and k-intends to dissect these tweets. Different calculations that we didn't utilize could turn out to be more significant. Since we just looked profoundly into content information, additionally research could demonstrate that different calculations are better for different kinds of information. We investigated our outcomes utilizing two perception instruments, Gephi and Wordle. There is still much to be done in this perspective. All things considered we would perform Singular Value decomposition on our agreement network before running k-implies. Along these lines commotion would be evacuated and the bunching would be more solid. For those intrigued by further investigation along the lines of our contextual analysis, a whiz augmentation is play out the examination progressively in order to see how specie subjects develop with time.