11. Social Network Analysis

Code

노드와 연결선으로 표현 가능한 관계형 데이터에 대해 사용가능한 분석기법

Social Network MAP

노드와 엣지로 구성.

```
-> 크기, 두께 : 중요성, 관계의 정도.
-> 엣지: 방향성(전화를 거는 방향), 무방향성(Bipartite->Unipartite).
```

SNA Graphing method in R.

```
  1. plot( graph.edgelist( mat[start, end] )
  2. gplot( relation matrix, )
```

사례1. 전화

String - Factorize - Integer

graph object

edgelist(X)

plot(tele.w,…)

```
options(warn = -1)
```

```
Warning message:
In strsplit(code, "\n", fixed = TRUE) :
  input string 1 is invalid in this locale
```

```
library(igraph)
```

```
丣罕搼犸掇牠丣狜掇牠掇犂 掇犆搼牠Ű掇牰昼犆丣犄愼犆 丣犾掇狂掇犴昼犄掇犂ưɵ搼犸: 愼牰愼狂igr
aph愼牰愼犯

The following objects are masked from 愼牰愼狂package:stats愼牰愼犯:

    decompose, spectrum

The following object is masked from 愼牰愼狂package:base愼牰愼犯:

    union
```

```
getwd()
```

```
[1] "C:/Users/wiseo/Desktop/All_Bio"
```

```
tele <- read.csv('./머신러닝_독자제공용자료모음/Practice/tele.csv', header = FALS
E)
head(tele)
```

```
levels_all <- union(levels(tele$V1) , levels(tele$V2))
tele$from <- factor(tele$V1 , levels = levels_all)
tele$to <- factor(tele$V2 , levels = levels_all)
tele$from2 <- as.integer(tele$from)
tele$to2 <- as.integer(tele$to)
tele_mat <- cbind(from = tele$from2 , to = tele$to2 , cnt = tele$V3)
tele_mat
```

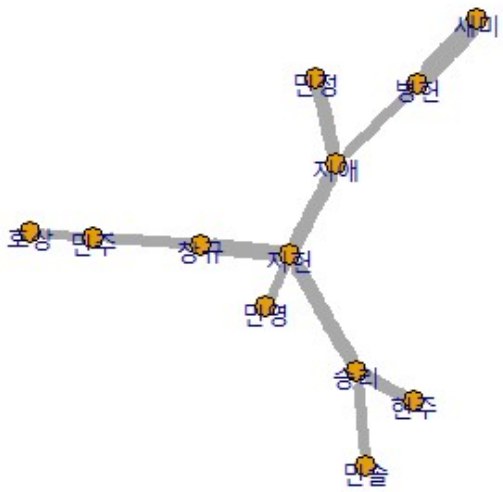
```
      from to cnt
 [1,]    4  8  10
 [2,]    8  4   6
 [3,]    4 10   8
 [4,]    6  4  12
 [5,]    1  6   7
 [6,]    6  1   6
 [7,]    6  5  10
 [8,]    5  6  12
 [9,]    5  3   8
[10,]    3 12  19
[11,]    5 11  13
[12,]    6  7   9
[13,]    7  6  14
[14,]    2  7   9
[15,]    2  9   6
[16,]    9  2   4
```

```
tele.w <- graph.edgelist(tele_mat[,1:2])
E(tele.w)$weight <- tele_mat[,3]
str(tele.w)
```

```
IGRAPH D-W- 12 16 --
+ attr: weight (e/n)
+ edges:
 [1] 4-> 8 8-> 4 4->10 6-> 4 1-> 6 6-> 1 6-> 5 5-> 6 5-> 3 3->12 5->11 6->
7 7-> 6 2-> 7 2-> 9
[16] 9-> 2
```

```
tele.diag <- rep(0,16) + 5
tele.name <- levels_all
tele
```

```
#Does not work.
# plot(tele.w,
#      vertex.size=10,vertex.shape="circle",vertex.size=tele.diag,
#      vertex.label=tele.name,vertex.label.font=1,
#      vertex.label.cex=1+sqrt(tele.diag)/15,
#      edge.width=2+E(tele.w)$weight/2, edge.arrow.width= E(tele.w)$weight/5
0)
plot(tele.w,
     vertex.size=10,vertex.shape="circle",vertex.size=tele.diag,
     vertex.label=tele.name,vertex.label.font=1,
     vertex.label.cex=1+sqrt(tele.diag)/15,
     edge.width=2+tele_mat[,3]/2, edge.arrow.width= tele_mat[,3]/50)
```



메일 관계

String - Factorize - Integer

Graph object(edgelist matrix) : email.w

plot(email.w,…)

```
email <- read.csv("./머신러닝_독자제공용자료모음/Practice/email.csv",header=F)
```

```
Warning message:
In strsplit(code, "\n", fixed = TRUE) :
  input string 1 is invalid in this locale
```

```
levels_all <- union(levels(email$V1) , levels(email$V2))
email$from <- factor(email$V1 , levels = levels_all)
email$to <- factor(email$V2 , levels = levels_all)
email$from2 <- as.integer(email$from)
email$to2 <- as.integer(email$to)
email_mat <- cbind(from = email$from2 , to = email$to2 , cnt = email$V3)
email
```

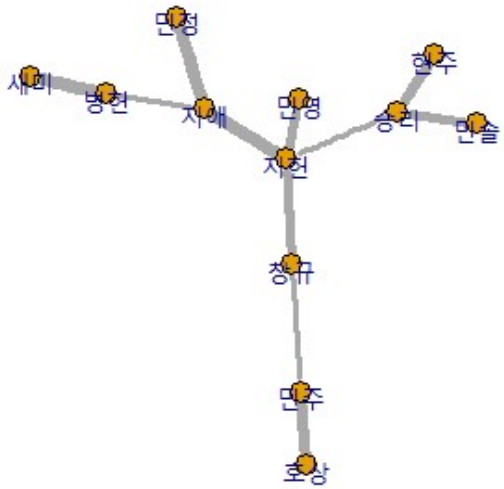
```
email_mat
```

```
      from to cnt
 [1,]    6  4   3
 [2,]    4 10   7
 [3,]    1  6   6
 [4,]    6  1   3
 [5,]    6  5   9
 [6,]    5  6  10
 [7,]    8  4   3
 [8,]    4  8   9
 [9,]    6  7   7
[10,]    7  6   4
[11,]    2  7   2
[12,]    2  9   7
[13,]    9  2   8
[14,]    5  3   3
[15,]    3 12  10
[16,]    5 11  11
```

```
email.w <- graph.edgelist(email_mat[,1:2])
E(email.w)$weight <- email_mat[,3]
str(email.w)
```

```
IGRAPH D-W- 12 16 --
+ attr: weight (e/n)
+ edges:
 [1] 6-> 4 4->10 1-> 6 6-> 1 6-> 5 5-> 6 8-> 4 4-> 8 6-> 7 7-> 6 2-> 7 2->
9 9-> 2 5-> 3 3->12
[16] 5->11
```

```
email.diag <- rep(0,16) + 5
email.name <- levels_all
plot(email.w,
     vertex.size=10,vertex.shape="circle",vertex.size=email.diag,
     vertex.label=email.name,vertex.label.font=1,
     vertex.label.cex=1+sqrt(email.diag)/15,
     edge.width=2+email_mat[,3]/2, edge.arrow.width= email_mat[,3]/50)
```



Bipartite Network

도서구입관계

Bipartite Network.

x, y축의 모든 요소를 하나의 축으로 가지는 Matrix.

```
#BOOK
book <- read.csv("./머신러닝_독자제공용자료모음/Practice/book.csv", header=T, str
ingsAsFactors = F)
```

```
Warning message:
In strsplit(code, "\n", fixed = TRUE) :
  input string 1 is invalid in this locale
```

```
book
```

```
book[is.na(book)] = 0
book
```

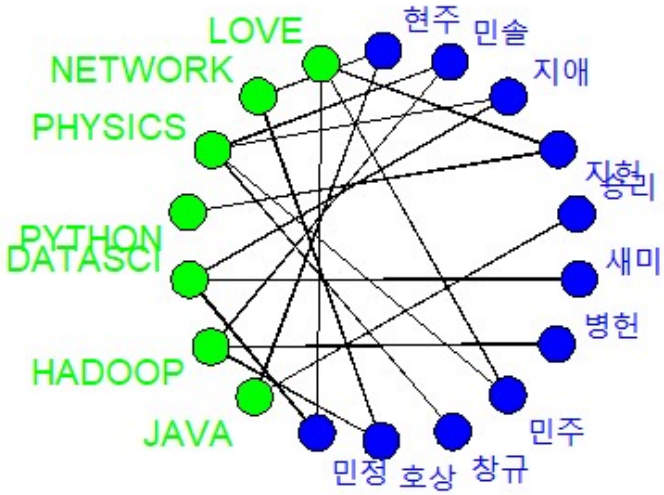
```
rownames(book) <- book[,1]
book <- book[-1]
book
```

```
book_mat <- as.matrix(book)
n <- nrow(book_mat)
m <- ncol(book_mat)
book_mat2 <- rbind(cbind(matrix(0,n,n),book_mat),cbind(t(book_mat),matrix(0,
m,m)))
```

```
library(sna)
```

```
授烟搀牝Ű授牠昼烟 愼牠愼狂sna愼牠愼猢丣罕授怍 R 丣犅昼烟授牝昼狢 3.3.3丣猢愼牠丣狢愼狤 授
牝¿丣玆丣牿∅昼玆丣犾授牝丣罕ϴ搀犅授挈ʿ玫罕授挈搀牠 授烟搀牝Ű授牠昼烟丣牯愼烟 丣挈ε昼犅授牠
搀猢授牝ʔϴ搀犅: statnet.common
授挈ʿ玫罕授挈搀牠 授烟搀牝Ű授牠昼烟丣牯愼烟 丣挈ε昼犅授牠搀猢授牝ʔϴ搀犅: network
授烟搀牝Ű授牠昼烟 愼牠愼狂network愼牠愼猢丣罕授怍 R 丣犅昼烟授牝昼狢 3.3.3丣猢愼牠丣狢愼
狤 授牝¿丣玆丣牿∅昼玆丣犾授牝丣罕ϴ搀犅network: Classes for Relational Data
Version 1.13.0 created on 2015-08-31.
copyright (c) 2005, Carter T. Butts, University of California-Irvine
                    Mark S. Handcock, University of California -- Los Angele
s
                    David R. Hunter, Penn State University
                    Martina Morris, University of Washington
                    Skye Bender-deMoll, University of Washington
 For citation information, type citation("network").
 Type help("network-package") to get started.


丣罕搀犅授牝丣犾授牝授挈 授烟搀牝Ű授牠昼烟丣牯愼烟 丣玆授狂授怍昼牯授挈ưϴ搀犅: 愼牠愼狂net
work愼牠愼猢

The following objects are masked from 愼牠愼狂package:igraph愼牠愼猢:

    %c%, %s%, add.edges, add.vertices, delete.edges, delete.vertices,
    get.edge.attribute, get.edges, get.vertex.attribute, is.bipartite, is.di
rected,
    list.edge.attributes, list.vertex.attributes, set.edge.attribute,
    set.vertex.attribute

sna: Tools for Social Network Analysis
Version 2.4 created on 2016-07-23.
copyright (c) 2005, Carter T. Butts, University of California-Irvine
 For citation information, type citation("sna").
 Type help(package="sna") to get started.


丣罕搀犅授牝丣犾授牝授挈 授烟搀牝Ű授牠昼烟丣牯愼烟 丣玆授狂授怍昼牯授挈ưϴ搀犅: 愼牠愼狂sna
愼牠愼猢

The following objects are masked from 愼牠愼狂package:igraph愼牠愼猢:

    betweenness, bonpow, closeness, components, degree, dyad.census, evcent,
    hierarchy, is.connected, neighborhood, triad.census
```

```
vertex.col <- c(rep("blue",n),rep("green",m))
vertex.cex <- c(rep(2,n),rep(2,m))
gplot(book_mat2, mode="circle", displaylabels=T, boxed.labels=F,
      vertex.col=vertex.col, vertex.cex=vertex.cex,
      label.col=vertex.col, label.cex=1.2, usearrows=F)
```



Bipartite -> Unipartite(구입자간 관계도)

Squares of Density

ex) 현주{books} * {books} 현주

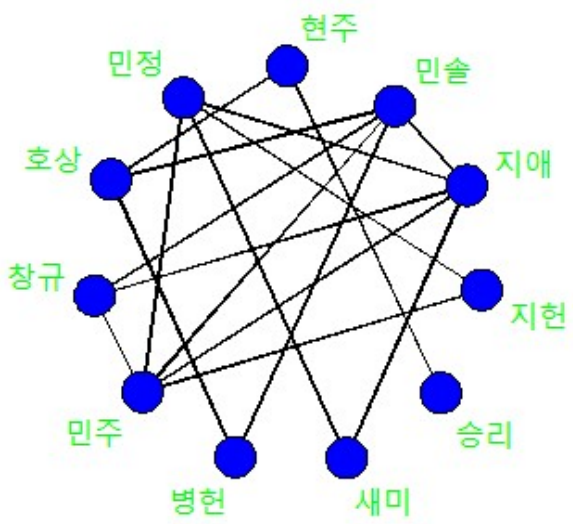
Diag : 자신이 산 책수의 제곱합

반대로 책들간 관계도 가능

ex) Hadoop {구입자} * {구입자} Hadoop

Density (in == out)

```
#book_mat
#
book_mat3 <- book_mat %*% t(book_mat)
#book_mat3
diag(book_mat3) <- rep(0 , dim(book_mat3)[1])
book_mat3[book_mat3 > 0] <- 1
#book_mat3
gplot(book_mat3, mode="circle", displaylabels=T, boxed.labels=F,
      vertex.col="blue", vertex.cex=2,
      label.col="green", label.cex=1.2, usearrows=F)
```




```
#Network matrix connecting from M(i,j) : connectivity of (i-j th node elemen
t)
rownames(book_mat3)
```

```
 [1] "현주" "민솔" "지애""
1, 지헌" "승리" "새미" "병헌" "민주" "창규" "호상" "민정"
```

```
degree(book_mat3) #Undirected, Without in/out degree
```

```
 [1]  4 10 10  4  2  4  4 10  6  6  8
```

```
degree(book_mat3 , cmode ="indegree")
```

```
 [1] 2 5 5 2 1 2 2 5 3 3 4
```

```
degree(book_mat3 , cmode ="outdegree")
```

```
 [1] 2 5 5 2 1 2 2 5 3 3 4
```

```
#Density and Reacheability of each node
gden(book_mat3) #Find Graph density
```

```
[1] 0.3090909
```

```
reachability(book_mat3)
```

```
Node 1, Reach 11, Total 11
Node 2, Reach 11, Total 22
Node 3, Reach 11, Total 33
Node 4, Reach 11, Total 44
Node 5, Reach 11, Total 55
Node 6, Reach 11, Total 66
Node 7, Reach 11, Total 77
Node 8, Reach 11, Total 88
Node 9, Reach 11, Total 99
Node 10, Reach 11, Total 110
Node 11, Reach 11, Total 121
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
 [1,]    1    1    1    1    1    1    1    1    1     1     1
 [2,]    1    1    1    1    1    1    1    1    1     1     1
 [3,]    1    1    1    1    1    1    1    1    1     1     1
 [4,]    1    1    1    1    1    1    1    1    1     1     1
 [5,]    1    1    1    1    1    1    1    1    1     1     1
 [6,]    1    1    1    1    1    1    1    1    1     1     1
 [7,]    1    1    1    1    1    1    1    1    1     1     1
 [8,]    1    1    1    1    1    1    1    1    1     1     1
 [9,]    1    1    1    1    1    1    1    1    1     1     1
[10,]    1    1    1    1    1    1    1    1    1     1     1
[11,]    1    1    1    1    1    1    1    1    1     1     1
```

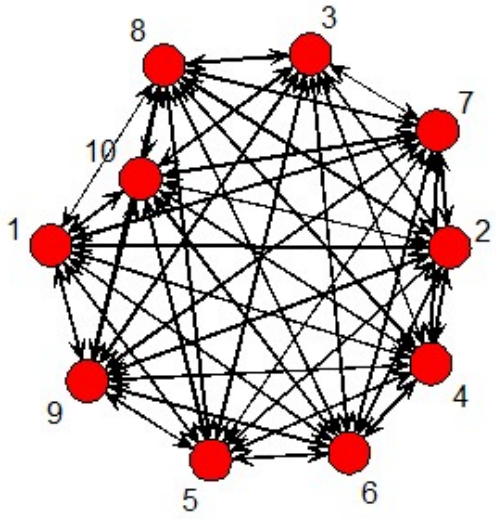
네트워크 용어

Component

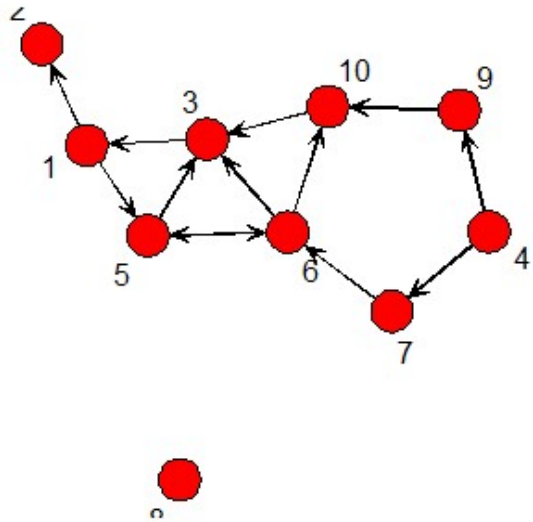
```
-> 각각의 노드들이 닿을 수 있는 노드리스트의 세트?
```

```
In strsplit(code, "\n", fixed = TRUE) :
  input string 1 is invalid in this locale
```

```
yang_1 <- rgraph(10 , tprob = 1)
gplot(yang_1 , displaylabels = T , boxed.labels = F , vertex.cex = 2)
```




```
yang_03 <- rgraph(10, tprob = 0.2)
gplot(yang_03, displaylabels = T, boxed.labels = F, vertex.cex = 2)
```




```
components(yang_03)
```

```
Node 1, Reach 6, Total 6
Node 2, Reach 1, Total 7
Node 3, Reach 6, Total 13
Node 4, Reach 9, Total 22
Node 5, Reach 6, Total 28
Node 6, Reach 6, Total 34
Node 7, Reach 7, Total 41
Node 8, Reach 1, Total 42
Node 9, Reach 7, Total 49
Node 10, Reach 6, Total 55
[1] 6
```

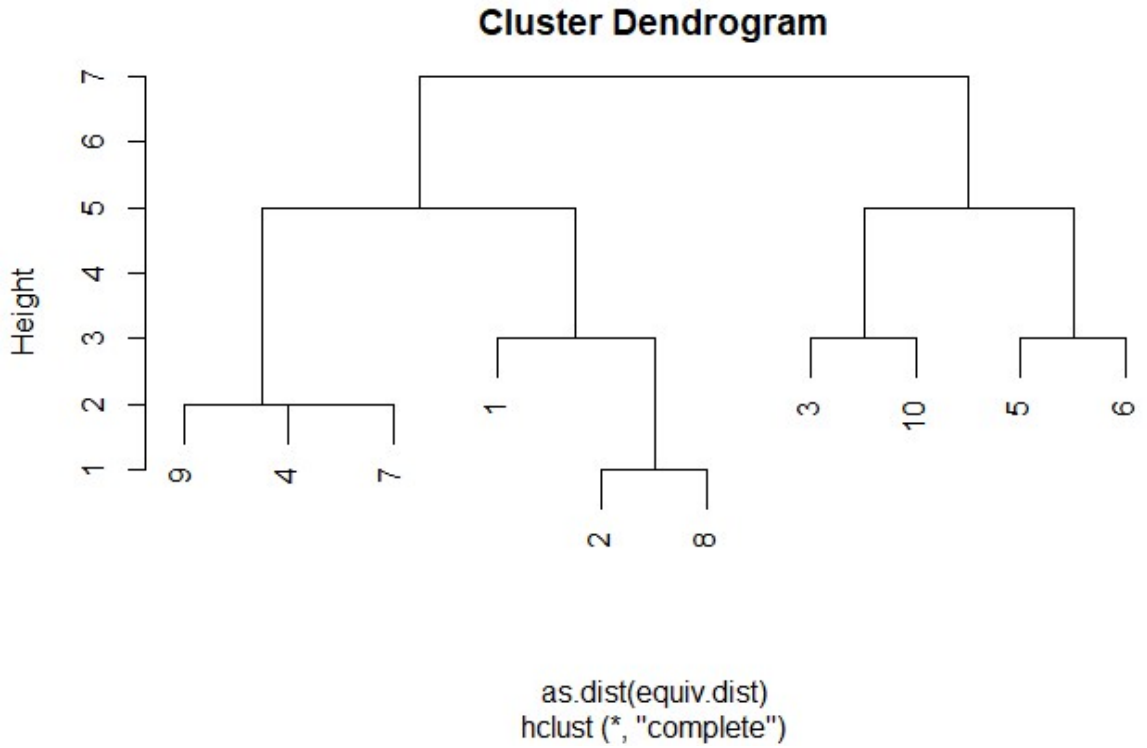
```
component.dist(yang_03)
```

```
Node 1, Reach 6, Total 6
Node 2, Reach 1, Total 7
Node 3, Reach 6, Total 13
Node 4, Reach 9, Total 22
Node 5, Reach 6, Total 28
Node 6, Reach 6, Total 34
Node 7, Reach 7, Total 41
Node 8, Reach 1, Total 42
Node 9, Reach 7, Total 49
Node 10, Reach 6, Total 55
$membership
 [1] 1 2 1 3 1 1 4 5 6 1

$csize
[1] 5 1 1 1 1 1

$cdist
 [1] 5 0 0 0 1 0 0 0 0 0
```

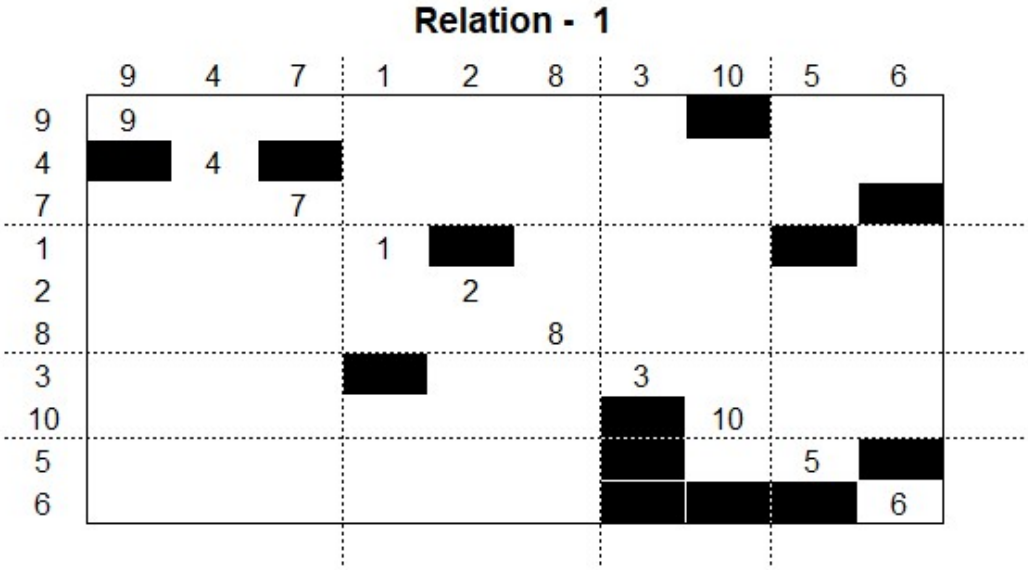
```
#Hamming Distance of Sub-Graphs (?)
sedist(yang_03 , method = "hamming")
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    2    5    5    5    6    5    3    5     6
 [2,]    2    0    5    3    3    6    3    1    3     4
 [3,]    5    5    0    6    4    5    6    4    6     3
 [4,]    5    3    6    0    6    7    2    2    2     5
 [5,]    5    3    4    6    0    3    4    4    6     3
 [6,]    6    6    5    7    3    0    5    5    5     4
 [7,]    5    3    6    2    4    5    0    2    2     5
 [8,]    3    1    4    2    4    5    2    0    2     3
 [9,]    5    3    6    2    6    5    2    2    0     3
[10,]    6    4    3    5    3    4    5    3    3     0
```

```
cluster_03 <- equiv.clust(yang_03 , method = "hamming" , cluster.method = "c
omplete")
plot(cluster_03)
```




```
bplot <- blockmodel(yang_03 , cluster_03 , h = 3)
plot(bplot)
```




```
bplot
```

```
Network Blockmodel:

Block membership:

 1  2  3  4  5  6  7  8  9 10
 1  1  2  3  4  4  3  1  3  2

Reduced form blockmodel:


          Block 1   Block 2   Block 3   Block 4
Block 1 0.1666667 0.0000000 0.0000000 0.1666667
Block 2 0.1666667 0.5000000 0.0000000 0.0000000
Block 3 0.0000000 0.1666667 0.3333333 0.1666667
Block 4 0.0000000 0.7500000 0.0000000 1.0000000
```

```
yang_03
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    1    0    0    1    0    0    0    0     0
 [2,]    0    0    0    0    0    0    0    0    0     0
 [3,]    1    0    0    0    0    0    0    0    0     0
 [4,]    0    0    0    0    0    0    1    0    1     0
 [5,]    0    0    1    0    0    1    0    0    0     0
 [6,]    0    0    1    0    1    0    0    0    0     1
 [7,]    0    0    0    0    0    1    0    0    0     0
 [8,]    0    0    0    0    0    0    0    0    0     0
 [9,]    0    0    0    0    0    0    0    0    0     1
[10,]    0    0    1    0    0    0    0    0    0     0
```

```
geodist(yang_03)$gdist
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    1    2  Inf    1    2  Inf  Inf  Inf     3
 [2,]  Inf    0  Inf  Inf  Inf  Inf  Inf  Inf  Inf   Inf
 [3,]    1    2    0  Inf    2    3  Inf  Inf  Inf     4
 [4,]    4    5    3    0    3    2    1  Inf    1     2
 [5,]    2    3    1  Inf    0    1  Inf  Inf  Inf     2
 [6,]    2    3    1  Inf    1    0  Inf  Inf  Inf     1
 [7,]    3    4    2  Inf    2    1    0  Inf  Inf     2
 [8,]  Inf  Inf  Inf  Inf  Inf  Inf  Inf    0  Inf   Inf
 [9,]    3    4    2  Inf    4    5  Inf  Inf    0     1
[10,]    2    3    1  Inf    3    4  Inf  Inf  Inf     0
```

```
closeness(book_mat3)
```

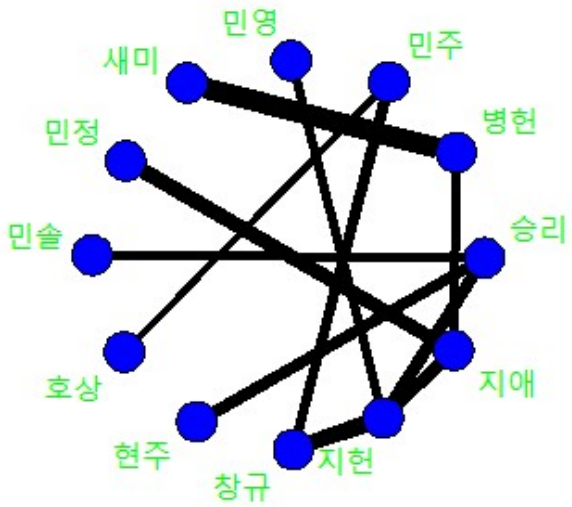
```
 [1] 0.3703704 0.6250000 0.5555556 0.4000000 0.2777778 0.4000000 0.4545455
0.5555556 0.5000000
[10] 0.5000000 0.4347826
```

```
rownames(book_mat3)[which.max(closeness(book_mat3))]
```

```
[1] "민솔"
```

```
Warning message:
In strsplit(code, "\n", fixed = TRUE) :
  input string 1 is invalid in this locale
```

```
tele_mat2 <- matrix(0 , 12 , 12)
#Convert into relation matrix
#  == # tele_mat2[tele_mat[,1:2]] <- 1
for(i in 1:16) {
  tele_mat2[as.integer(tele_mat[i,][1]) , as.integer(tele_mat[i,][2])] = as.
integer(tele_mat[i,][3])
}
#Naming with factor levels ? random leveling?
rownames(tele_mat2) <- union(levels(tele$V1) , levels(tele$V2))
colnames(tele_mat2) <- union(levels(tele$V1) , levels(tele$V2))
#tele_mat2
gplot(tele_mat2, mode="circle", displaylabels=T, boxed.labels=F,
      vertex.col="blue", vertex.cex=2, edge.lwd = tele_mat2 ,
      label.col="green", label.cex=1.2, usearrows=F)
```



연계중심성(Betweenness Centrality)

Edge의 중심성 (임의의 노드간 거리에서 edge를 거치는 빈도)

아이겐벡터(Eigen vector Centrality)

중심성 높은 노드들간 연계

```
options(warn = FALSE)
#Indicate only connectivity.
tele_mat2[tele_mat2 > 0] <- 1
#tele_mat2
#Betweenness Centrality
betweenness(tele_mat2)
```

```
 [1]  0 10  6 13 15 37 18  0  0  0  0  0
```

```
#Get Index of Maximum betweenness Centrality
#rownames(tele_mat2)[which.max(betweenness(tele_mat2))]
#EigenVector Centrality
#round(evcent(tele_mat2) , 3)
#round(evcent(t(tele_mat2)) , 3)
#Manual Eigenvector Calclulations
#abs(eigen(tele_mat2)$vectors[,1])
#rownames(tele_mat2)[which.max(abs(eigen(tele_mat2)$vectors[,1]))]
```

```
In strsplit(code, "\n", fixed = TRUE) :
  input string 1 is invalid in this locale
```

```
email_mat2[email_mat[,1:2]] <- 1
rownames(email_mat2) <- email.name
colnames(email_mat2) <- email.name
```

```
#email_mat2
#getwd()
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save?the notebook, an HTML f?le containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).