

# 2015 KRUG 컨퍼런스 SNA

Chad (Chungil Chae)

June 24, 2015

## 소개



Ph.D Candidate (ABD) Dual title degree in Workforce Education and Development & Comparative International Education

[chadchae@gmail.com](mailto:chadchae@gmail.com) / [chadchae.github.io](http://chadchae.github.io)

<http://rpubs.com/chadchae/krugdocument2015>

<http://rpubs.com/chadchae/krugpresentation2015>

## 목차

- Social Network Analysis 사회연결망분석
  - Social Network Analysis 란?
  - 관계, 구조, 트랜잭션
  - SNA Software
  - SNA R package
- SNA with R
  - 네트워크 자료 구조
    - 매트릭스
    - 리스트
    - R 에서의 네트워크 데이터 생성
    - R 에서의 네트워크 데이터 로드
- SNA with R
  - 네트워크레벨
  - 개별노드레벨
- Gephi 를 사용한 시각화
- 소셜네트워크의 통계적 추정
  - Social Influence Model
  - Social Selection Model
- Exponential Random Graph Model
  - Exponential Random Graph Model 란?
  - 가정
  - 수식
  - Configuration
  - 모델추정
  - 결과해석
  - Goodness of Model Fit
- Statnet Web
- Reference

## Social Network Analysis 란?

- 사회현상을 이루는 객체간 상호작용의 연결성을 구조 또는 교환으로 이해하고 분석하는 방법.
- 소셜네트워크분석과 다른 분석들간의 가장 큰 차이점은 관심대상의 관계의 개념과 정보를 포함하고 있다는 점.

## 관계, 구조, 트랜잭션

- 관계: 전체 네트워크의 연결성을 규정하는 기본 연결규칙
  - 친구관계
  - 부부관계
  - 조언관계
  - 협력관계
- 구조: 네트워크에 참여자의 증가와 관계의 다양성, 연결에서 나타나는 관계들의 형태
- 트랜잭션: 어떤 작업에 대한 일련의 연속

## 주요 소셜네트워크 분석 소프트웨어

- UCINET
- PAJCK
- 넷마이너
- SOCNETV
- Gephi
- 파이톤
- R

[위키피디아 - 소셜네트워크 분석 소프트웨어](#)

## SNA R Package

- sna package
- igraph package
- network package
- statnet package
- ergm package

R에서의 SNA 분석은 반복적인 네트워크 생성-분석-재생성 과정의 처리와 네트워크의 계량적인 지수들의 도출과 통계적 방법론을 적용할때 장점을 보인다.

## SNA with R

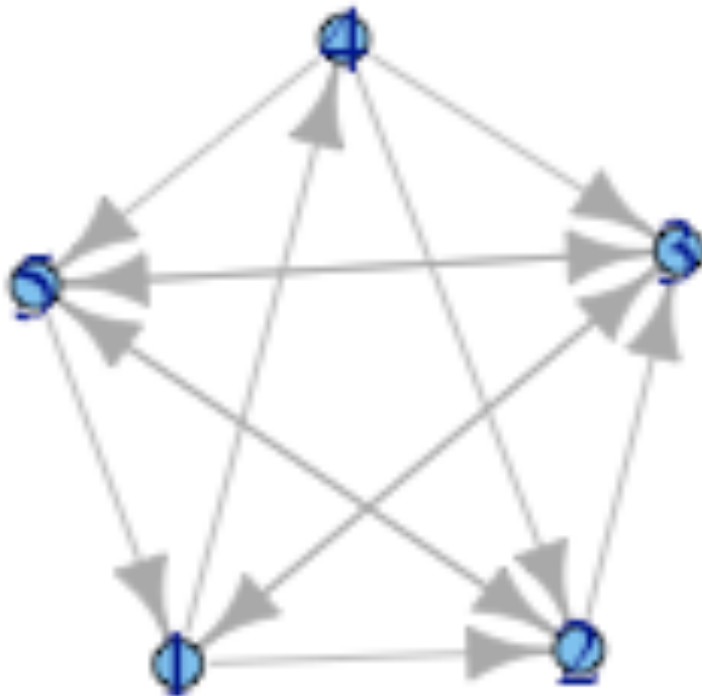
### 자료구조

#### 메트릭스

연결망은 그래프로 나타내어 질 수 있으며 행렬은 그래프를 이루는 가장 기본적인 형식

```
## sna: Tools for Social Network Analysis
## Version 2.3-2 created on 2014-01-13.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
## For citation information, type citation("sna").
## Type help(package="sna") to get started.

##      [,1] [,2] [,3] [,4] [,5]
## [1,]  0   1   1   1   0
## [2,]  0   0   1   0   1
## [3,]  1   0   0   0   1
## [4,]  0   1   1   0   1
## [5,]  1   1   1   0   0
```



# SNA with R

## 자료구조

### 리스트

- 1:1 의 관계로 구성된 자료의 리스트
- 많은 양의 자료를 수집, 처리에 용이함
- 통상 소스 - 타겟의 형식

```
##
## Attaching package: 'igraph'
##
## The following objects are masked from 'package:sna':
##
##   %c%, betweenness, bonpow, closeness, degree, dyad.census,
##   evcent, hierarchy, is.connected, neighborhood, triad.census
##
## Edge sequence:
##
## [1] 2 -- 1
## [2] 3 -- 1
## [3] 3 -- 2
## [4] 4 -- 2
## [5] 5 -- 4
## [6] 6 -- 4
## [7] 7 -- 4
## [8] 6 -- 5
## [9] 7 -- 6
```

## 방향성 없는 네트워크

엣지리스트 형태로 자료를 입력

```
g <- graph.formula (1-2, 1-3, 2-3, 2-4, 4-5, 4-5, 4-6, 4-7, 5-6, 6-7)
```

노드의 확인

**V**(g)

```
## Vertex sequence:
```

```
## [1] "1" "2" "3" "4" "5" "6" "7"
```

엣지의 확인

**E**(g)

```
## Edge sequence:
```

```
##
```

```
## [1] 2 -- 1
```

```
## [2] 3 -- 1
```

```
## [3] 3 -- 2
```

```
## [4] 4 -- 2
```

```
## [5] 5 -- 4
```

```
## [6] 6 -- 4
```

```
## [7] 7 -- 4
```

```
## [8] 6 -- 5
```

```
## [9] 7 -- 6
```

네트워크 자료구조

**str**(g)

```
## IGRAPH UN-- 7 9 --
```

```
## + attr: name (v/c)
```

```
## + edges (vertex names):
```

```
## [1] 1--2 1--3 2--3 2--4 4--5 4--6 4--7 5--6 6--7
```

인접행렬을 확인

**get.adjacency**(g)

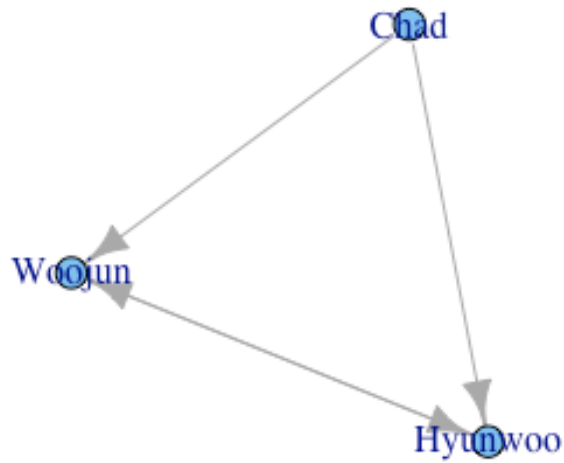
```
## 7 x 7 sparse Matrix of class "dgCMatrix"
```

```
##  1 2 3 4 5 6 7
```

```
## 1 . 1 1 . . . .  
## 2 1 . 1 1 . . .  
## 3 1 1 . . . . .  
## 4 . 1 . . 1 1 1  
## 5 . . . 1 . 1 .  
## 6 . . . 1 1 . 1  
## 7 . . . 1 . 1 .
```

## 방향성 있는 네트워크의 경우

```
dg <- graph.formula(Chad->Woojun, Chad->Hyunwoo, Woojun->Hyunwoo)
V(dg)$name
## [1] "Chad" "Woojun" "Hyunwoo"
plot(dg)
```





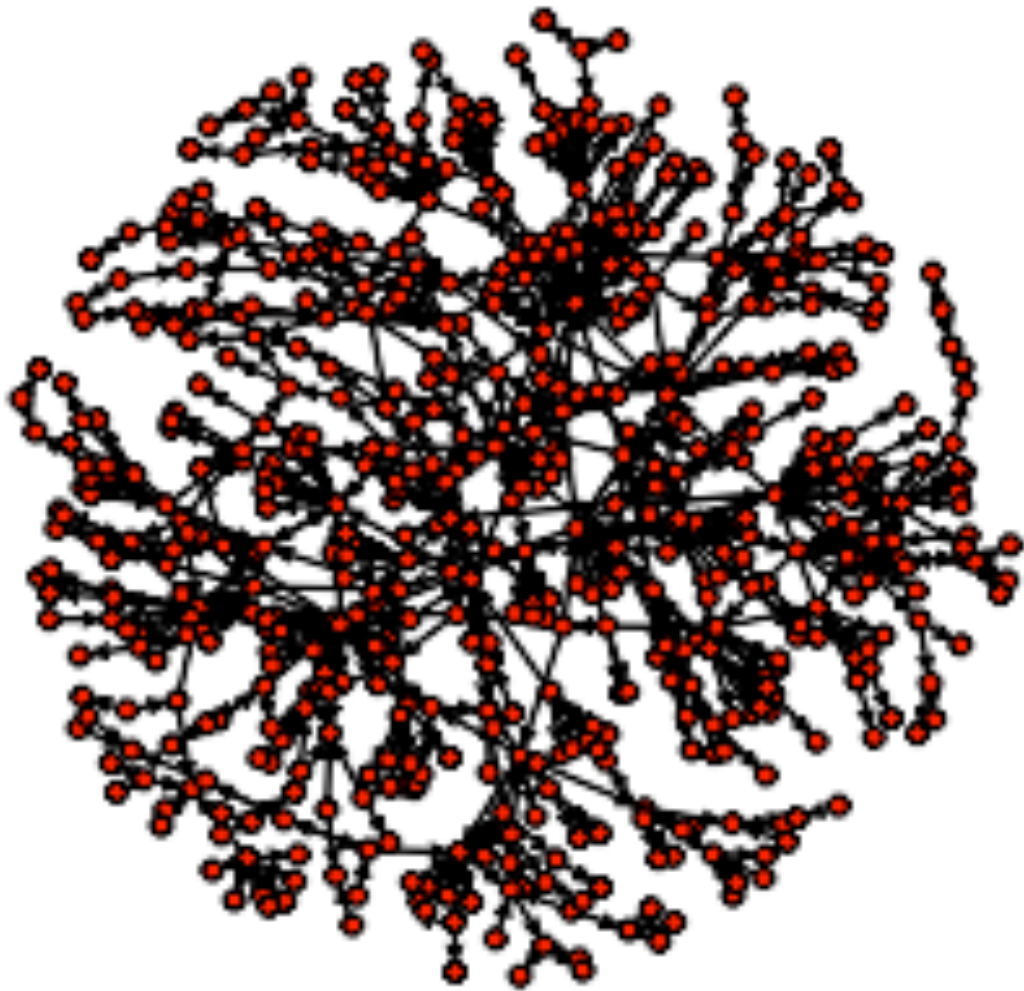
## R 에서의 네트워크 데이터 로드

```
require(network)

edges <- read.csv("Dataset/edges.csv")
nodes <- read.csv("Dataset/nodes.csv")
id <- seq(1,593,1)
nodes<-cbind(id, nodes)
edges2<-merge(nodes[,1:2], edges, by.x = "au_id", by.y="Target")
names(edges2)[1]<-"alter_id"
names(edges2)[2]<-"alter_R_id"
edges3<- merge(nodes[,1:2], edges2, by.x = "au_id", by.y="Source")
names(edges3)[1]<-"ego_id"
names(edges3)[2]<-"ego_R_id"
net<-network(edges3[,c("ego_R_id", "alter_R_id")])

## Network attributes:
##  vertices = 593
##  directed = TRUE
##  hyper = FALSE
##  loops = FALSE
##  multiple = FALSE
##  bipartite = FALSE
##  total edges = 749
##  missing edges = 0
##  non-missing edges = 749
##  density = 0.002133563
##
## Vertex attributes:
##  vertex.names:
##  character valued attribute
##  593 valid vertex names
##
## No edge attributes
##
## Network edgelist matrix:
##      [,1] [,2]
## [1,]  1 501
## [2,]  3 467
```

```
plot(net)
```



## 분석

- 네트워크레벨
  - 네트워크 크기
  - 네트워크 밀도
  - 네트워크 중심성 경향 (Centralization)
- 개별노드레벨
  - 노드의 중심도

## 네트워크 레벨

```
## Loading required package: intergraph
```

노드 사이즈

```
length(V(unet))
```

```
## [1] 593
```

엣지 사이즈

```
length(E(unet))
```

```
## [1] 748
```

평균패스길이

```
average.path.length(unet, directed=FALSE)
```

```
## [1] 6.71653
```

네트워크 밀도

```
graph.density(unet)
```

```
## [1] 0.002130714
```

네트워크 중심성 경향

```
centralization.evcent(unet)$centralization
```

```
## [1] 0.9784051
```

네트워크 지름 (가장 긴 path), 어떤 네트워크에서 노드들의 연결중 가장 긴 연결이 일렬로 연결된 세 노드가 있다면 네트워크 지름은 2 그 네트워크 지름을 구성하는 노드는 3

*# 지름*

`diameter(unet)`

## [1] 17

*# 지름을 이루는 노드들의 조합*

`get.diameter(unet)`

## [1] 396 419 195 199 404 287 436 241 511 452 324 369 512 575 429 588 591

## [18] 579

평균거리 Average Path Length

`average.path.length(unet, directed=FALSE)`

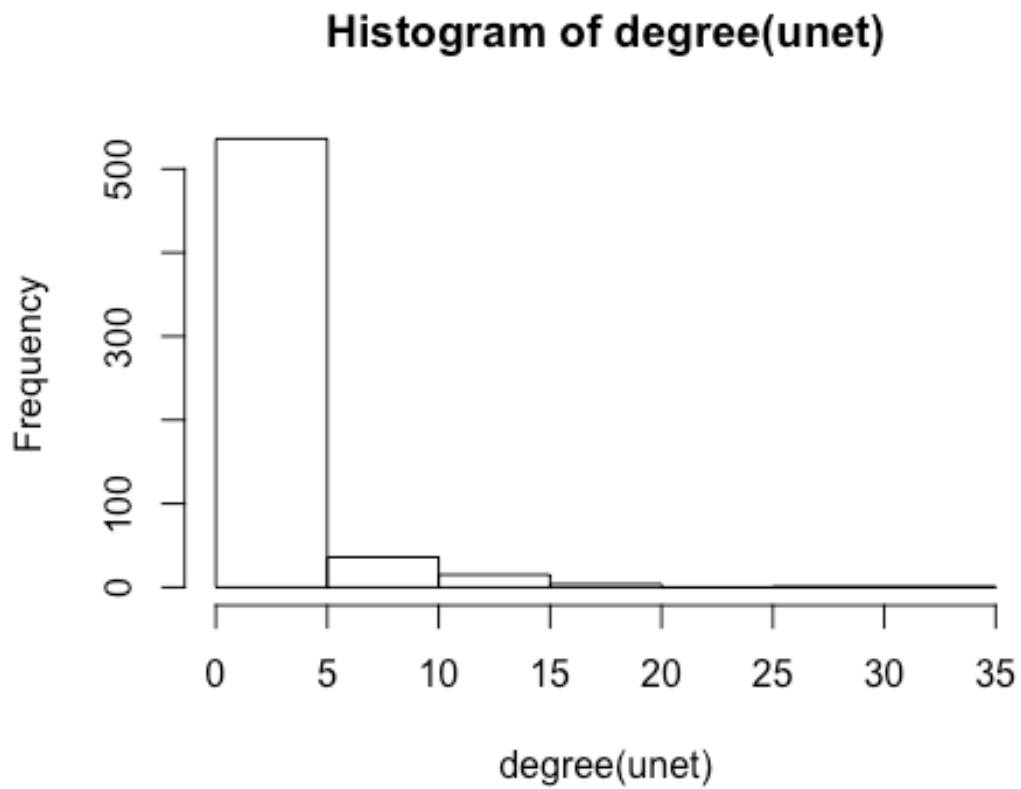
## [1] 6.71653

## 개별 노드 레벨

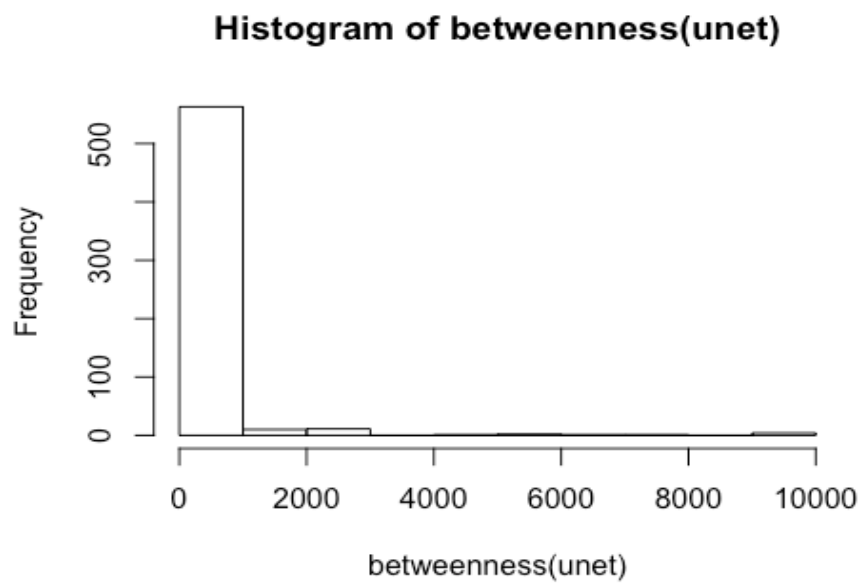
`degree(unet)`

```
## [1] 1 3 2 1 8 4 1 3 3 1 2 4 12 1 1 1 1 1 1 2 2 1 1
## [24] 5 1 3 1 1 1 1 1 1 1 2 1 1 2 1 1 1 2 1 2 8 2 2
## [47] 1 1 1 1 1 5 1 2 1 2 1 1 1 1 1 1 3 4 1 1 5 1 1
## [70] 8 1 1 1 1 1 1 2 1 2 1 3 3 1 1 1 1 2 2 1 2 2 1
## [93] 1 1 1 3 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 10 6
## [116] 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 6 1 1 1 1 1 1
## [139] 1 1 2 1 1 2 2 1 1 1 2 1 1 1 1 4 1 1 1 6 5 4 3
## [162] 2 1 2 1 2 2 2 1 1 5 3 2 1 1 4 1 1 1 2 1 1 1 1
## [185] 2 1 1 2 2 1 1 1 3 1 2 1 2 2 14 1 3 1 3 1 1 1 1
## [208] 1 1 2 1 4 1 1 1 8 1 3 1 1 1 3 1 2 10 1 2 1 1 1
## [231] 1 1 1 2 3 1 1 2 1 1 9 1 1 1 1 1 1 1 2 1 1 1 1
## [254] 3 1 1 1 4 1 2 1 3 1 1 3 1 1 1 1 1 1 1 1 18 1 3
## [277] 1 1 1 1 1 1 1 1 1 2 10 1 1 1 3 1 2 1 2 1 3 1 1
## [300] 2 2 4 2 11 1 1 2 1 1 4 1 6 1 1 1 1 7 1 1 3 1 1
## [323] 1 5 1 1 2 1 1 1 2 1 5 1 1 2 2 1 1 1 1 1 1 2 3
## [346] 1 2 1 1 1 1 1 1 1 1 4 1 2 1 2 2 1 1 1 3 4 2 1
```

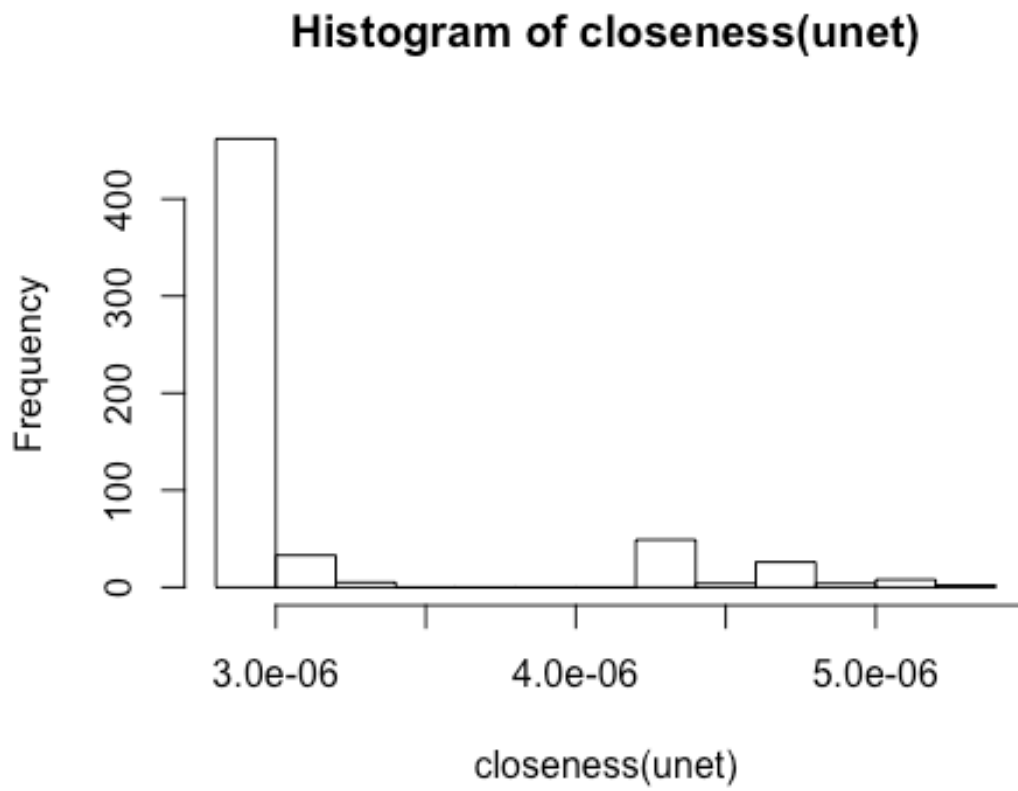
```
hist(degree(unet))
```



```
hist(betweenness(unet))
```



```
hist(closeness(unet))
```



## 기타 분석

노드와 엣지의 카운트

```
ecount(unet)
```

```
## [1] 748
```

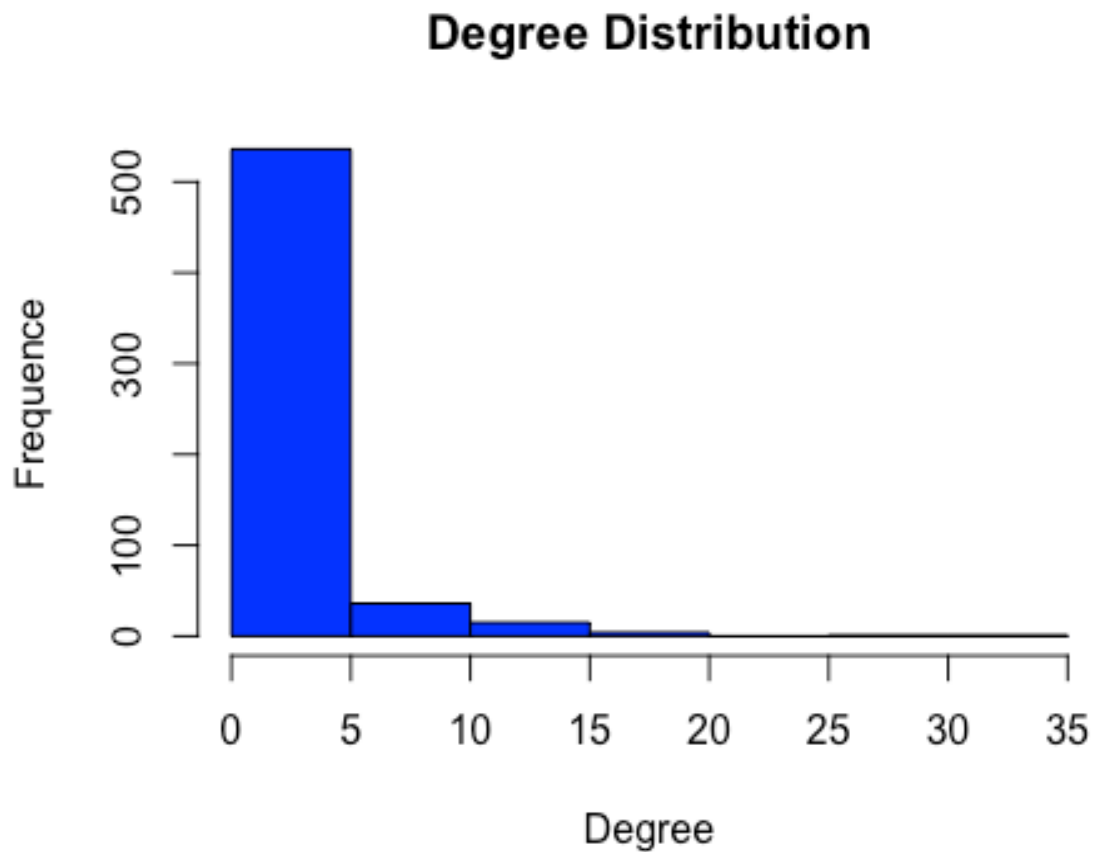
```
vcount(unet)
```

```
## [1] 593
```

로그 - 로그 디그리 분포

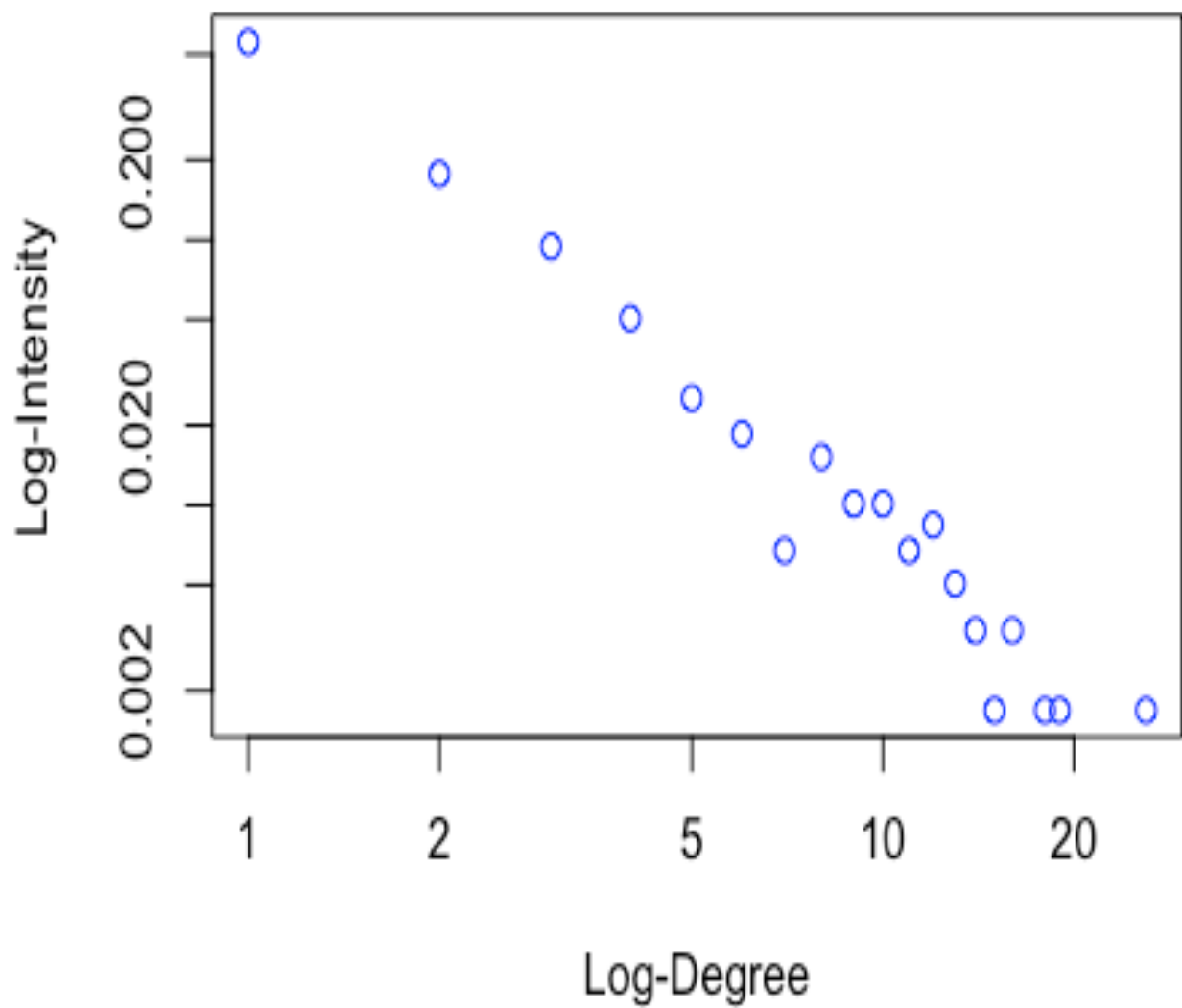
```
d.unet <-degree(unet)
```

```
hist(d.unet, col="blue", xlab = "Degree", ylab="Frequency", main="Degree Distributi  
on")
```



```
dd.unet <- degree.distribution(unet)
d<-1:max(d.unet)-1
ind <- (dd.unet != 0)
plot(d[ind], dd.unet[ind], log="xy", col = "blue",
      xlab=c("Log-Degree"), ylab=c("Log-Intensity"),
      main = "Log-Log Degree Distribution")
```

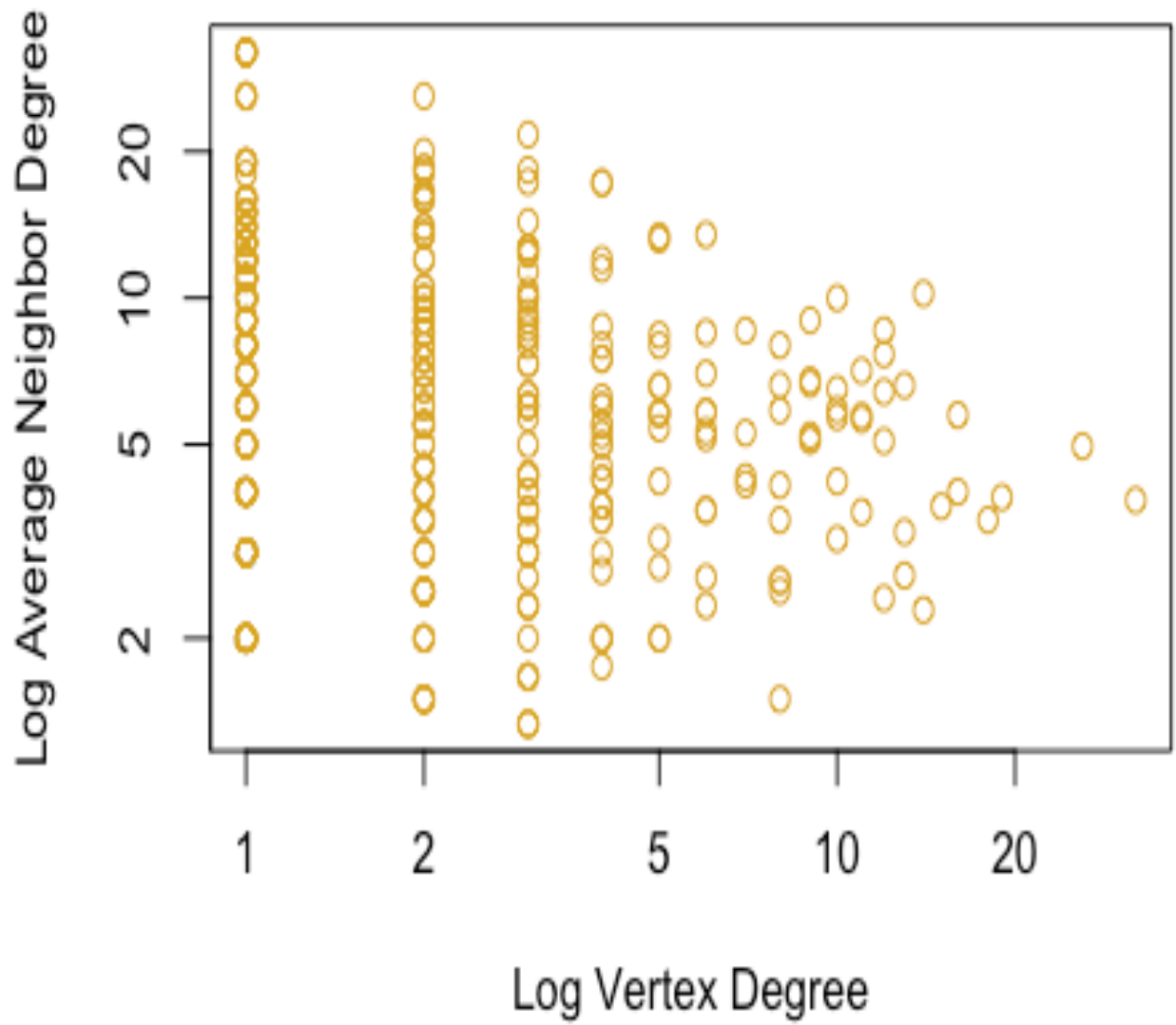
## Log-Log Degree Distribution





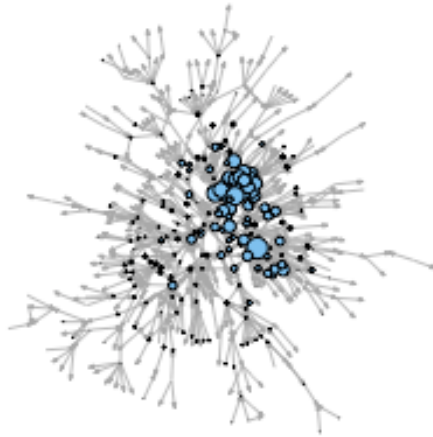
로그 평균 이웃 디그리

```
snet <- simplify(unet)
a.nn.deg.unet <- graph.knn(snet, V(unet))$knn
plot(d.unet, a.nn.deg.unet, log="xy",
     col="goldenrod", xlab=c("Log Vertex Degree"),
     ylab=c("Log Average Neighbor Degree"))
```



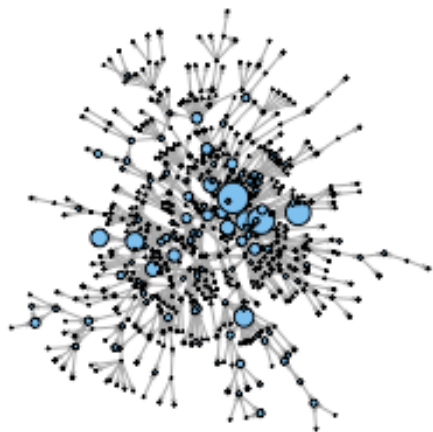
HUB / Authority 점수가 노드 크기에 반영된 그래프

```
l<-layout.kamada.kawai(unet)
plot(unet, layout=l, edge.width=0.8, edge.arrow.size=0.1,
     main ="HUB", vertex.label="", vertex.size=10 * sqrt (hub.score(unet)$vector))
```



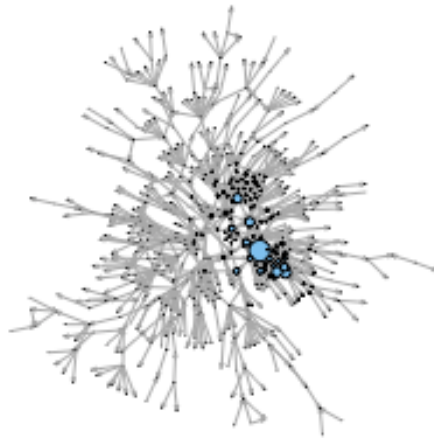
페이지 랭크 점수가 노드의 크기에 반영된 그래프

```
pr <- page.rank(unet)$vector
plot(unet, layout=l, vertex.size=pr*1000, vertex.label=NA, edge.arrow.size=.1)
```

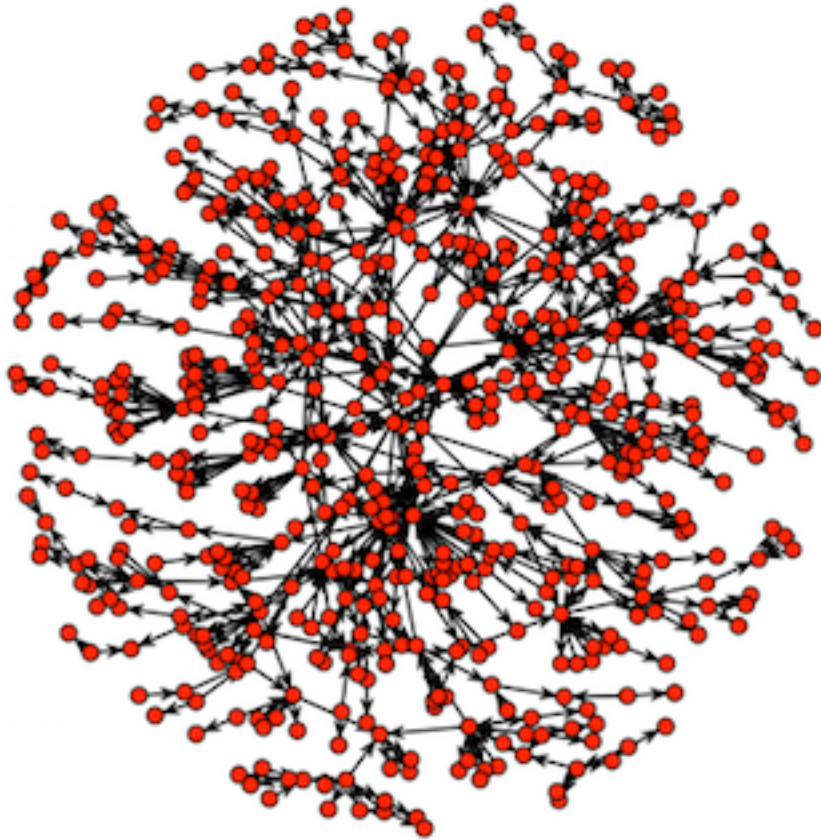


Eigenvector 중심도가 노드 크기에 반영된 그래프

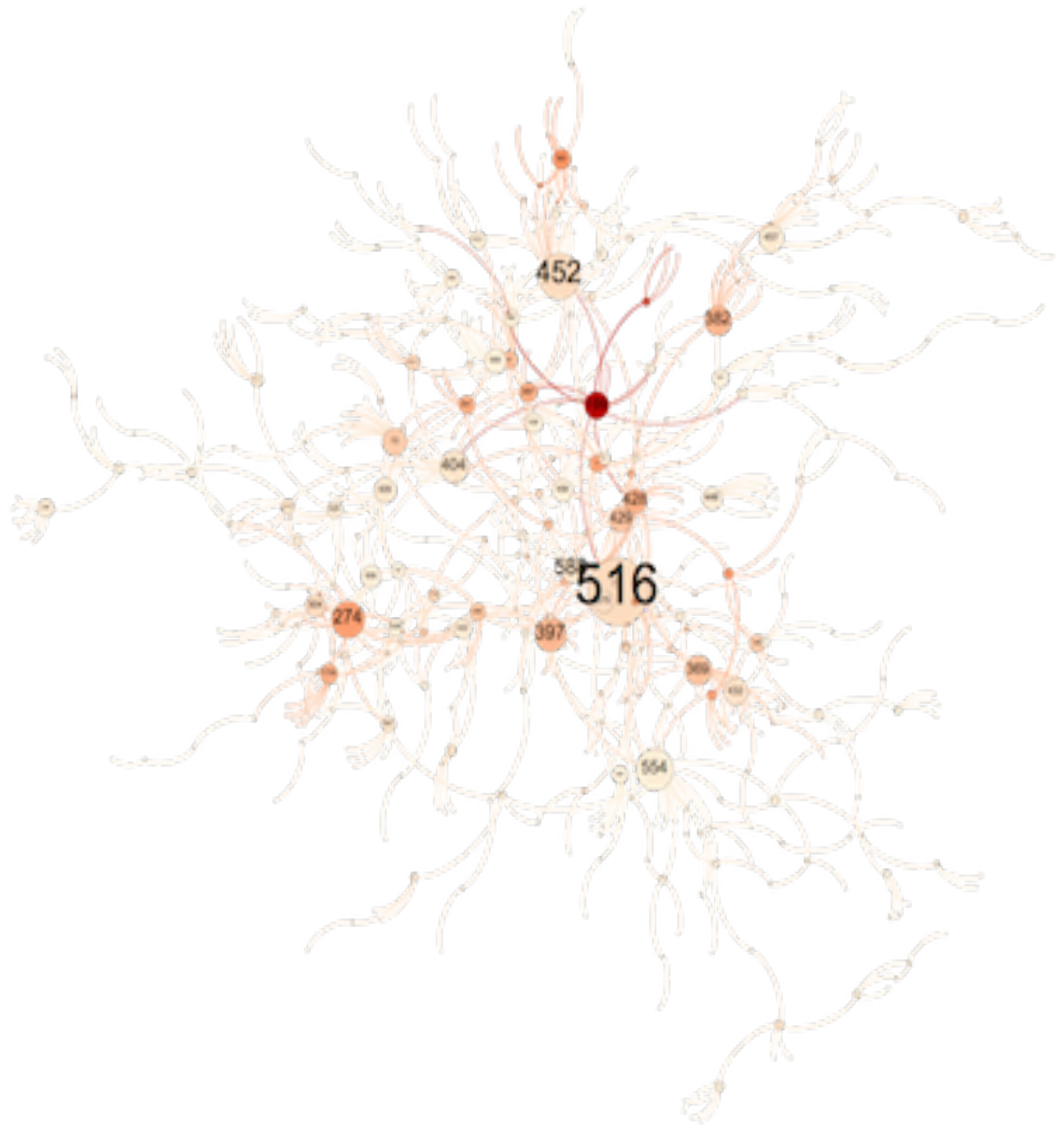
```
ec <- evcent(unet)$vector  
plot(unet, layout=l, vertex.size=ec*10, vertex.label=NA, edge.arrow.size=.1)
```



## R 에서의 기본 네트워크 시각화



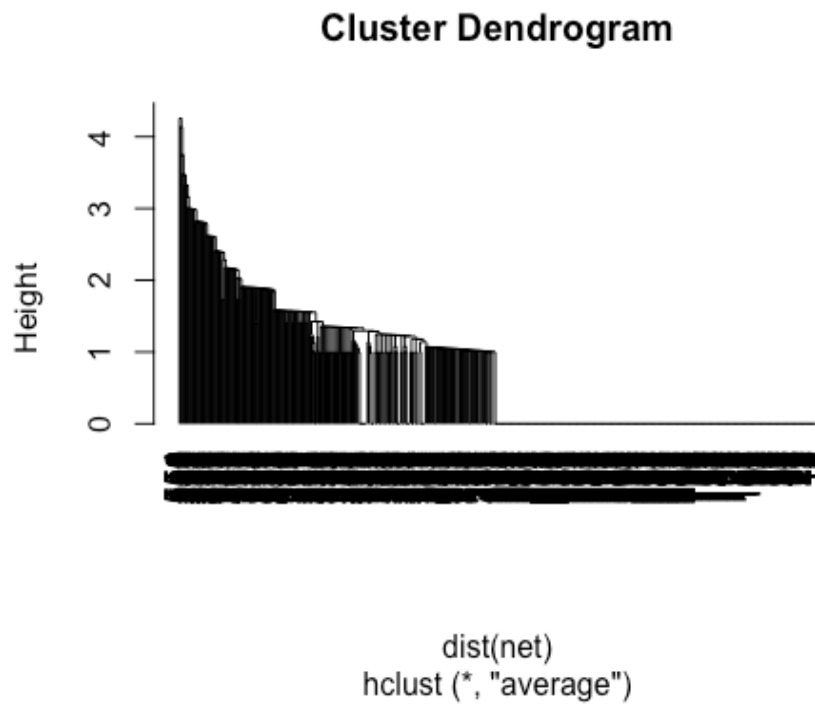
- Gephi 를 사용한 네트워크 시각화



## 클러스터링과 Heatmap

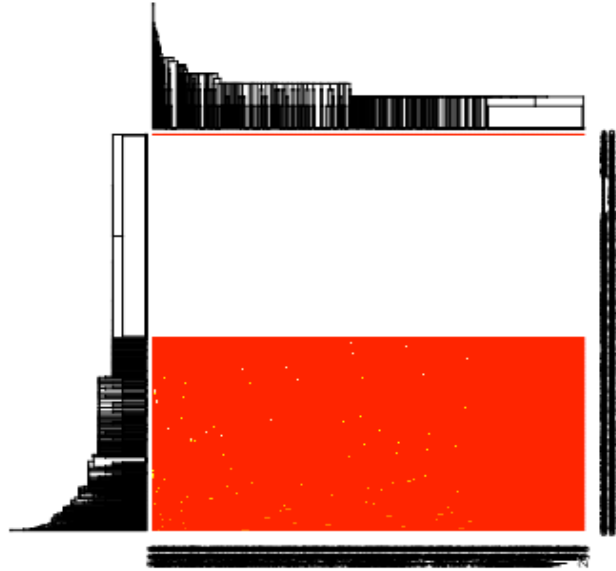
네트워크 클러스터링

```
#Clustering  
hc <- hclust(dist(net), method="ave")  
plot(hc, hang=-1)
```



Heatmap

```
#heatmap  
heatmap(as.matrix(net))
```



## 소셜네트워크 모델

### Social Influence Model

네트워크 변수들을 독립변수에 넣어 구조적 위치와 종속변수와 연관이 있는지를 확인하는 모델

- 종속변수 = 독립변수 1 + 독립변수 2 + 네트워크변수 1 + 네트워크변수 2

### Social Selection Model

네트워크 변수가 종속변수가 되어 관측된 네트워크에 영향 또는 연관이 있는 독립변수들의 설명에 주목

- 네트워크변수 = 독립변수 + 독립변수 2 + Configuration1 + Configuration2

# Exponential Random Graph

## Exponential Random Graph Model 이란?

"Exponential random graph models (ERGMs) 들은 네트워크 구조에 대한 통계적 모형이며, 네트워크 연결들이 패턴화되는지를 추정하는것을 가능하게 한다.

ERGMs 은 네트워크의 타이들을 기반으로 한 모델로서 어떻게 그리고 왜 소셜네트워크의 연결들이 일어나는지를 설명하는 모델이다" (Lusher, Koshinen, and Robins, 2013, p.9)

Lusher, Koshinen, Robins. (2013), Exponential Random Graph Models for Social Networks : Theory, Methods, and Applications

"네트워크 로지스틱 회귀모형에서는 설명 네트워크에 조건화 하여 종속 네트워크 Y 의 모든 연결성이 독립이라고 가정된다. 그러나 노드연결은 독립이라고 가정된다. 그러나 사회 네트워크에서 노드 연결은 상호 종속성을 보일 것으로 기대된다. 예컨데 추이성(transitivity)과 상호성(reciprocity) 같은 것이 있을 수 있다. 이에 따라 최근 개발된 네트워크 모형에서는 이런 네트워크의 기본 속성이 반영되었다"(허명희, 2012, p.70)

허명희 (2012), R 을 활용한 사회네트워크분석 입문, 자유아카데미

ERGM 모델은 네트워크내의 개개인의 결과를 예측하는 모델이 아니며, Social Selection 모델로서 네트워크 내의 타이들의 형성에 영향을 미치는 패턴을 추론하는 모델이다.

## 가정

- 소셜네트워크들은 국지적으로 나타난다
- 네트워크 tie 는 자기조직적일 뿐만 아니라 다른 행위자의 속성 또는 외부적 요소에 영향을 받는다
- 네트워크내의 패턴은 진행되고있는 구조적 과정의 증거로 볼 수 있다.
- 여러 프로세스들은 동시에 나타날 수 있다.
- 소셜 네트워크는 구조화 되어있다, 그러나 추론적(Stochastic)이다.

## 수식

$$\Pr(Y = y) = \left(\frac{1}{k}\right) \exp\left\{\sum A \eta_A g_A(y)\right\}$$



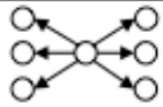





- $y$  는 관측된 네트워크
- $\eta_A$  는 Configuration A 에 대한 파라미터
- $g_A(y)$  는 Configuration A 에 대한 통계량
- $k$  는 모든 가능한 configuration  $\{\sum \eta_A g_A(y)\}$  을 합하여 계산된 normalizing factor

## Configurations

네트워크 연결은 중요한 패턴을 형성하는데 이런 연결로 생기는 주요 기본적 구조를 Configuration 이라고 한다. 그리고 이러한 Configuration 은 ERGM 모델 내에서 파라미터화 되어 있다.

*ERGM configurations tested for the co-authorship network in HRD*

Configuration	Shape	Explanation
Edges		It equals to the number of edges in the network.
Mutual		It equals to the number of pairs of authors $i$ and $j$ for which $(i \rightarrow j)$ and $(j \rightarrow i)$ both exist.
Out Star		The $k$ argument is a vector of distinct integers. This term adds one network statistic to the model for each element in $k$ .
Two Path		This term adds one statistic to the model, equal to the number of 2-paths in the network. For a directed network this is defined as a pair of edges $(i \rightarrow j), (j \rightarrow k)$ , where $j$ and $k$ must be distinct.
Cycle		The $k$ argument is a vector of distant integers. This term adds one network statistic to the model for each element in $k$ ; the $i$ th such statistic equals the number of cycles in the network with length exactly $k[i]$ . It captures a non-hierarchical exchange of co-authorship.
ESP (Edgewise shared partners)		This term adds one network statistic to the model for each element in $d$ where the $i$ th such statistic equals the number of edges in the network with exactly $d[i]$ shared partners. This captures ego's edge connection driven by connections between alters.

Specification of Exponential-Family Random Graph Models: Terms and Computational Aspects(<http://www.jstatsoft.org/v24/i04/paper>)

## 모델추정

```
require(ergm)

m6<-ergm(net ~ edges + mutual + ctriple)

## Starting maximum likelihood estimation via MCMLE:
## Iteration 1 of at most 20:
## The log-likelihood improved by 4.787
## Iteration 2 of at most 20:
## The log-likelihood improved by 2.792
## Iteration 3 of at most 20:
## The log-likelihood improved by 1.669
## Step length converged once. Increasing MCMC sample size.
## Iteration 4 of at most 20:
## The log-likelihood improved by 0.4178
## Step length converged twice. Stopping.
##
## This model was fit using MCMC. To examine model diagnostics and check for de
generacy, use the mcmc.diagnostics() function.

# 모델결과
summary(m6)

##
## =====
## Summary of model fit
## =====
##
## Formula: net ~ edges + mutual + ctriple
##
## Iterations: 4 out of 20
##
## Monte Carlo MLE Results:
##      Estimate Std. Error MCMC % p-value
## edges  -6.3347   0.0408    0 <1e-04 ***
## mutual   4.2495   0.1589    1 <1e-04 ***
## ctriple  3.0904   0.2646    0 <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Null Deviance: 486667 on 351056 degrees of freedom
## Residual Deviance: 10367 on 351053 degrees of freedom
##
## AIC: 10373 BIC: 10405 (Smaller is better.)
```

### *# MCMC Diagnostics*

```
mcmc.diagnostics(m6)
```

```
## Sample statistics summary:
```

```
##
```

```
## Iterations = 16384:4209664
```

```
## Thinning interval = 1024
```

```
## Number of chains = 1
```

```
## Sample size per chain = 4096
```

```
##
```

```
## 1. Empirical mean and standard deviation for each variable,
## plus standard error of the mean:
```

```
##
```

```
##      Mean    SD Naive SE Time-series SE
## edges 17.418 29.917 0.46745    2.0331
## mutual  5.763  6.481 0.10127    0.9371
## ctriple 1.560  4.433 0.06927    0.3506
```

```
##
```

```
## 2. Quantiles for each variable:
```

```
##
```

```
##      2.5% 25% 50% 75% 97.5%
## edges  -42 -2 18 38 75
## mutual  -10  2  6 10 18
## ctriple  -6 -2  1  4 11
```

```
##
```

```
##
```

```
## Sample statistics cross-correlations:
```

```
##      edges  mutual  ctriple
## edges  1.0000000 0.4692517 0.4059864
## mutual  0.4692517 1.0000000 0.2224949
## ctriple 0.4059864 0.2224949 1.0000000
```

```
##
```

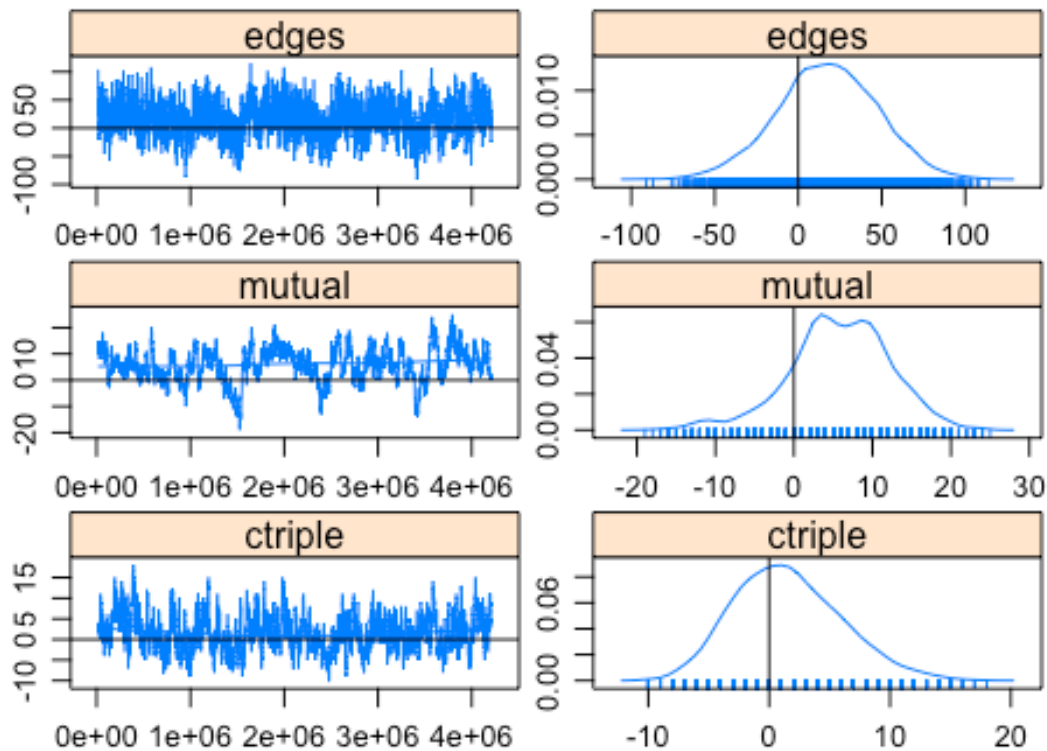
```
## Sample statistics auto-correlation:
```

```
## Chain 1
```

```
##      edges  mutual  ctriple
```

```
## Lag 0 1.0000000 1.0000000 1.0000000
## Lag 1024 0.6609287 0.9769092 0.9279571
## Lag 2048 0.4841307 0.9545249 0.8580338
## Lag 3072 0.3975131 0.9328383 0.7934274
## Lag 4096 0.3547699 0.9115949 0.7353201
## Lag 5120 0.3359119 0.8912221 0.6821174
##
## Sample statistics burn-in diagnostic (Geweke):
## Chain 1
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
## edges mutual ctriple
## 3.0076 0.4997 2.4376
##
## Individual P-values (lower = worse):
## edges mutual ctriple
## 0.002633476 0.617283119 0.014786584
## Joint P-value (lower = worse): 0.04937257 .
## Loading required namespace: latticeExtra
```

## Sample statistics



##

## Recent changes in the ergm estimation algorithm mean that these plots can no longer be used to ensure that the mean statistics from the model match the observed network statistics. For that functionality, please use the GOF command: `gof(ergmFit Object, GOF=~model)`.

## 결과해석

ERGM 모델은 네트워크 패턴의 강도나 방향을 나타내는 파라미터 추정치를 구한다. 파라미터 추정치의 Positive 값은 변수로 넣은 configuration 의 영향이 기대보다 많다는 것을 의미. 파라미터 값의 크기는 바로 다른 효과들과 비교될수 없고 ODD Ratio 형태로 비교 가능하다.

```
summary(m6)

##
## =====
## Summary of model fit
## =====
##
## Formula: net ~ edges + mutual + ctriple
##
## Iterations: 4 out of 20
##
## Monte Carlo MLE Results:
##      Estimate Std. Error MCMC % p-value
## edges  -6.3347   0.0408    0 <1e-04 ***
## mutual   4.2495   0.1589    1 <1e-04 ***
## ctriple  3.0904   0.2646    0 <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Null Deviance: 486667 on 351056 degrees of freedom
## Residual Deviance: 10367 on 351053 degrees of freedom
##
## AIC: 10373 BIC: 10405 (Smaller is better.)
## [1] 3.250471
```

트라이앵글형태의 연결보다는 밀착된 연결이 3 배 가량 연결 가능도가 높다.

## 모델평가 (Goodness of Model Fit)

```
m1_1.gof <- gof(m6~distance)
m1_2.gof <- gof(m6~espartners)
m1_3.gof <- gof(m6~idegree)
m1_4.gof <- gof(m6~odegree)
```

## Distance

m1\_1.gof

##

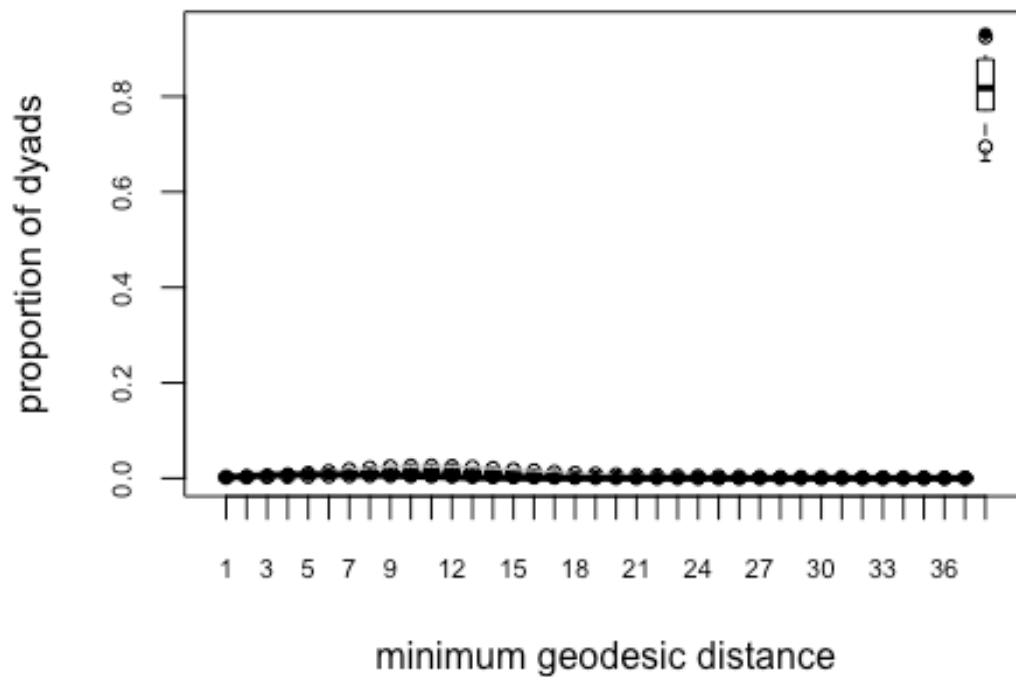
## Goodness-of-fit for minimum geodesic distance

##

##	obs	min	mean	max	MC p-value
## 1	748	692	760.61	858	0.62
## 2	1576	815	1017.79	1351	0.00
## 3	2260	980	1351.28	2168	0.00
## 4	2769	1081	1763.24	3361	0.04
## 5	2724	1183	2255.75	5020	0.44
## 6	2770	1243	2811.58	7167	0.86
## 7	2527	1293	3383.17	9490	0.68
## 8	2205	1318	3903.08	11598	0.34
## 9	1990	1334	4308.86	12865	0.18

[plot](#)(m1\_1.gof)

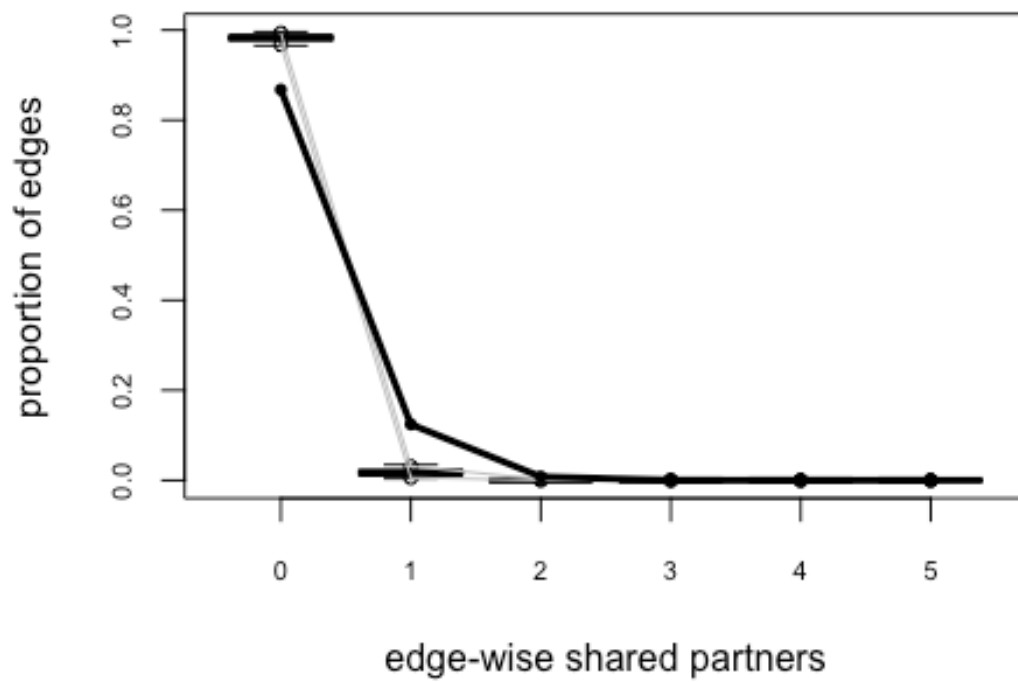
## Goodness-of-fit diagnostics



Edge wise shard partner

```
plot(m1_2.gof)
```

## Goodness-of-fit diagnostics

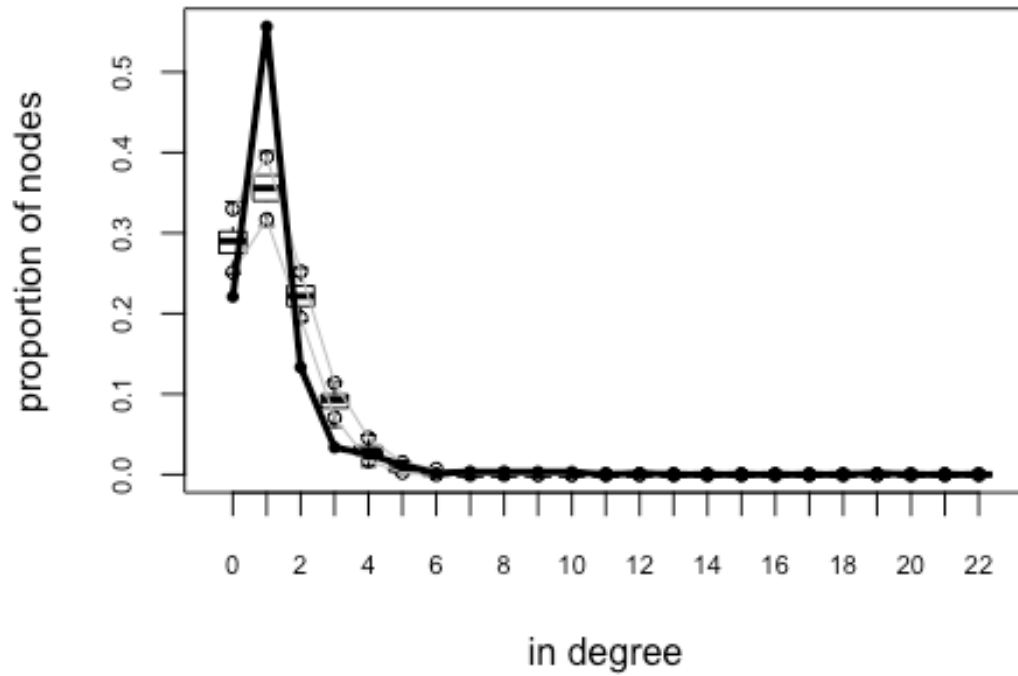


In-degree

```
plot(m1_3.gof)
```



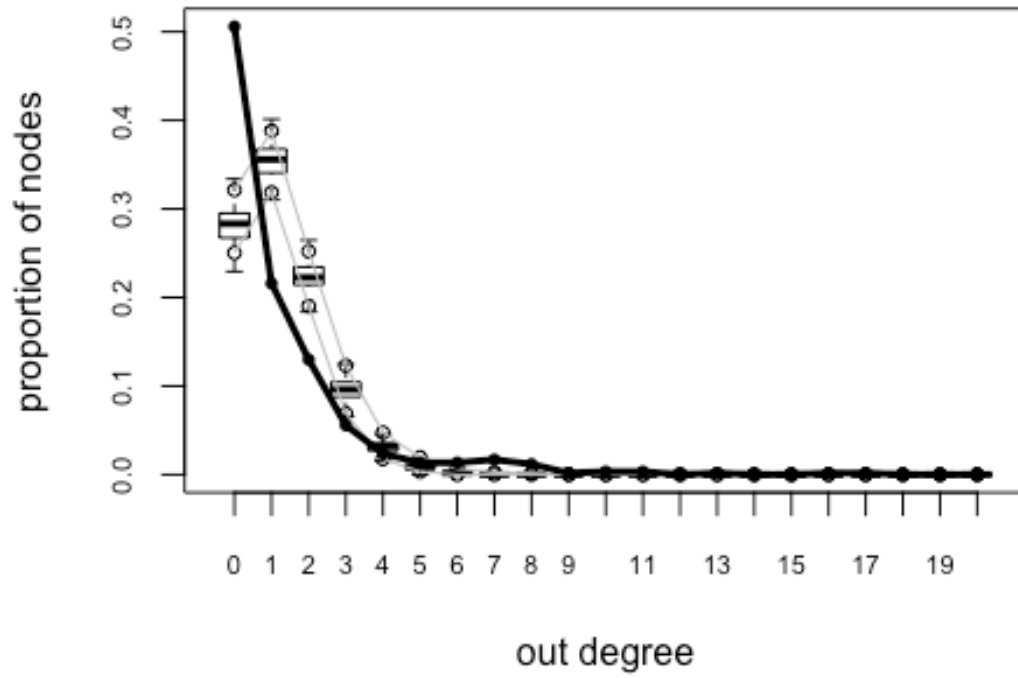
## Goodness-of-fit diagnostics



Out-degree

```
plot(m1_4.gof)
```

## Goodness-of-fit diagnostics



# Statenet Web

## Shiny 로 구현한 네트워크분석, ERGM 분석 툴 Statenet Web

statnetWeb

Data

Network Descriptives

Fit Model

MCMC Diagnostics

Goodness of Fit

Simulations

Help

About statnetWeb

Citing statnetWeb

Get Started

Welcome to the first interactive interface for the **ergm** package! **ergm** is part of the statnet network analysis software - a suite of packages written in R - and this GUI also includes some of the functionality from the associated packages **network** and **sna** . This web application is written with the Shiny framework from RStudio and development is via GitHub. More information on the statnet software, the ergm package, Shiny and our GitHub repository can be found in the resource links on the right.

This interface is useful for teachers and students of introductory network analysis, for newcomers to exponential random graphs models, and for experienced network modelers who want easier access to analysis results. If you are new to ergm, you may find it helpful to work through the [ergm tutorial](#) using this interface. Advanced users will still want to interact via the command line in order to access the full functionality of ergm.

A typical network analysis will move sequentially through the tabs at the top of the page. Click on the help icon at the top of any page for guidance.

Do you have comments/suggestions/complaints on this prototype app? Please share them with us. They are best submitted through our [GitHub site](#), or by email to the statnet\_help mailing list (see [Help](#) tab).

Resources

[About statnetWeb](#)

[About statnet](#)


Key background papers «

Tutorials and documentation «

[statnetWeb on GitHub](#)

[Shiny: a web application framework for R](#)

UNIVERSITY of WASHINGTON

 CENTER FOR STUDIES IN  
DEMOGRAPHY AND ECOLOGY



# Reference

## Books and Articles

- Lusher, Koshinen, Robins. (2013), Exponential Random Graph Models for Social Networks : Theory, Methods, and Applications
- 허명희 (2012), R 을 활용한 사회네트워크분석 입문, 자유아카데미
- Kolaczyk (2014), Statistical analysis of network data, Springer
- Kolaczyk and Csardi (2014), Statistical analysis of network data with R, Springer
- Standford SNA lab (2014), <http://sna.stanford.edu/>
- Morris, Handcock, Hunter (2008) Specification of Exponential-Family Random Graph Models: Terms and Computational Aspects
- Wasserman and Faust (2009), Social Network Analysis: Methods and Applications

## R and packages

- R Core Team (2015), R: A language and environment for statistical computing
- Carter T. Butts (2014). sna: Tools for Social Network Analysis. R package version 2.3-2. <http://CRAN.R-project.org/package=sna>
- Csardi G, Nepusz T: The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006. <http://igraph.org>
- Mark S. Handcock, David R. Hunter, Carter T. Butts, Steven M. Goodreau, and Martina Morris (2003). statnet: Software tools for the Statistical Modeling of Network Data. URL <http://statnetproject.org>
- Bojanowski, Michal (2015) intergraph: Coercion Routines for Network Data Objects. R package version 2.0-1. <http://mbojan.github.io/intergraph>
- Butts C (2008). "network: a Package for Managing Relational Data in R." *Journal of Statistical Software*, 24(2). .
- Emily Beylerian (2015). statnetWeb: A Graphical User Interface for Network Modeling with 'Statnet'. R package version 0.3.4.
- Bastian M., Heymann S., Jacomy M. (2009). Gephi: an open source software for exploring and manipulating networks. International AAAI Conference on Weblogs and Social Media

End

경청해 주셔서 감사합니다

발표 프리젠테이션은 다음의 링크를 통해서 접속할 수 있습니다.

<http://rpubs.com/chadchae/krugdocument2015>

<http://rpubs.com/chadchae/krugpresentation2015>

문의사항이 있으시다면

[chadchae@gmail.com](mailto:chadchae@gmail.com) / [chadchae.github.io](http://chadchae.github.io)

연락부탁드립니다.