

# 네트워크 분석

네트워크의 특성을 파악하기 위한 지표에는 각 노드가 어느 정도로 많은 관계를 맺고 있는 지를 나타내는 연결척도(degree), 한 점이 다른 모든 점들에 얼마나 가까운가를 나타내는 근접(Closeness), 연결망에서 한 노드가 다른 노드들 사이에 위치하는 정도를 나타내는 중개(betweenness)가 있다.

각 지표는 중심성(centrality)과 중심화(centralization) 관점으로 설명된다.

중심성: 네트워크를 구성하는 각각의 노드가 전체 연결망에서 어느 정도 중심에 위치하는 가를 나타내며, 연결 정도의 중심성, 근접 중심성, 중개 중심성이 있다.

중심화: 네트워크 전체 연결망의 형태가 어느 정도 중앙에 집중되어 있는지를 나타내는 지표로 연결 정도 중심화, 근접 중심화, 중개 중심화가 있다.

밀도: 중심화와 같이 네트워크의 전체 특성을 나타내는 지표로 전체 노드가 서로 얼마나 많은 관계를 맺고 있는가를 알려준다.

## #네트워크분석 start 형 제작

### #네트워크 분석을 위한 igraph 패키지 설치

```
install.packages("igraph")  
library(igraph)
```

### #방향성이 없고 edges와 노드(n)가 없는 igraph 생성

```
g_star<-graph(edges=NULL,n=NULL,directed=F)  
plot(g_star) #노드가 없어 그래프가 생성되지 않음.
```

### #그래프에 A노드 추가, 노드의 모양은 원, 크기는 30, 색은 노랑

```
g_star<-g_star+vertex("A",shape="circle",size=30, color="yellow")  
plot(g_star)
```

### #그래프에 B,C,D,E,F 노드 추가, 노드의 모양은 원, 크기는 30

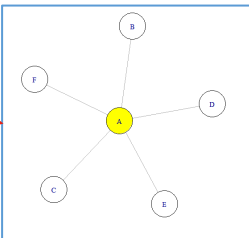
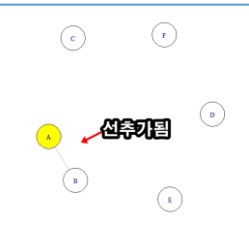
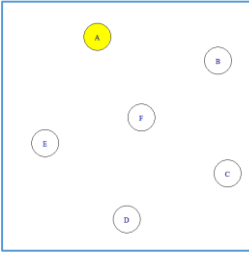
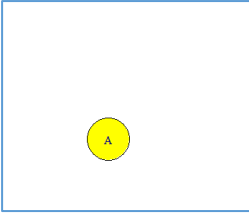
```
g_star<-g_star+vertex("B","C","D","E","F",shape="circle",size=30)  
plot(g_star)
```

### #그래프에 A와 B사이의 선추가

```
g_star<-g_star+edges("A","B")  
plot(g_star)
```

### # A-C, A-D, A-E, A-F 네트워크 생성

```
g_star<-g_star+edges("A","C","A","D","A","E","A","F")  
plot(g_star)
```



# 네트워크 분석

## #네트워크 분석을 위한 igraph 패키지 설치

```
install.packages("igraph")  
library(igraph)
```

## ##### Y자형 작성

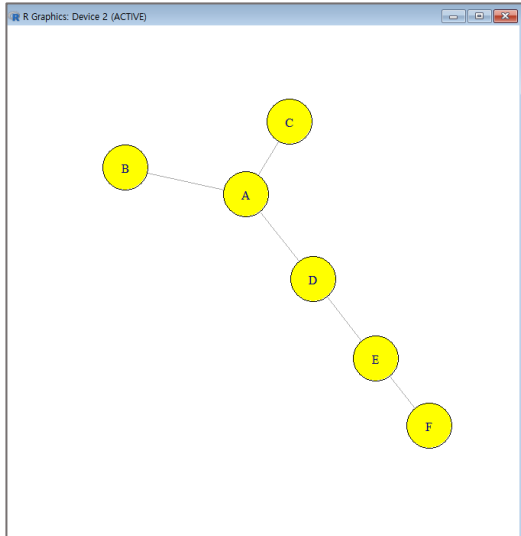
```
g_Y<-graph(edges=NULL,n=NULL,directed=F)  
g_Y<-g_Y+vertex("A","B","C","D","E","F",shape="circle",size=30, color="yellow")  
plot(g_Y)
```

## # A에 B,C,D연결, D에 E연결 (즉 A-B, A-C, A-D-E), E에 F연결(즉 A-D-E-F)

```
g_Y<-g_Y+edges("A","B","A","C","A","D","D","E","E","F")  
plot(g_Y)
```

vcount(g\_Y) #네트워크내의 노드수 (A,B,C,D,E,F), 6개

ecount(g\_Y) #노드간 연결된 에지의 총수 (a-b, a-c, a-d, d-e,e-f) 5개



## #네트워크 분석을 위한 igraph 패키지 설치

```
install.packages("igraph")  
library(igraph)
```

## ##### 원형

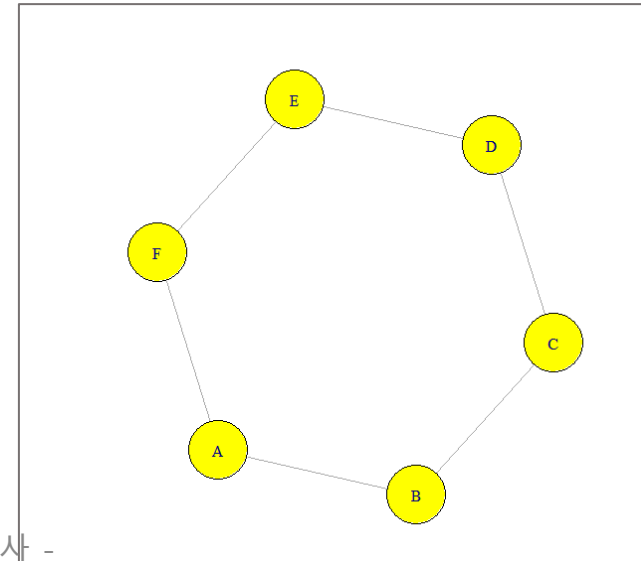
```
g_R<-graph(edges=NULL,n=NULL,directed=F)  
g_R<-g_L+vertex("A","B","C","D","E","F",shape="circle",size=30, color="yellow")  
plot(g_R)
```

## # A-B, B-C, C-D, D-E, E-F, F-A 연결

```
g_R<-g_R+edges("A","B","B","C","C","D","D","E","E","F","F","A")  
plot(g_R)
```

vcount(g\_R) #네트워크내의 노드수 (A,B,C,D,E,F), 6개

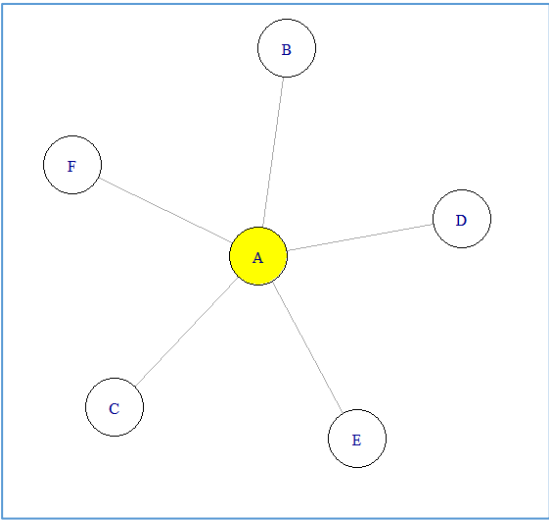
ecount(g\_R) #노드간 연결된 에지의 총수 (A-B, B-C, C-D, D-E, E-F, F-A) 6개



# 네트워크 구조별 연결정도

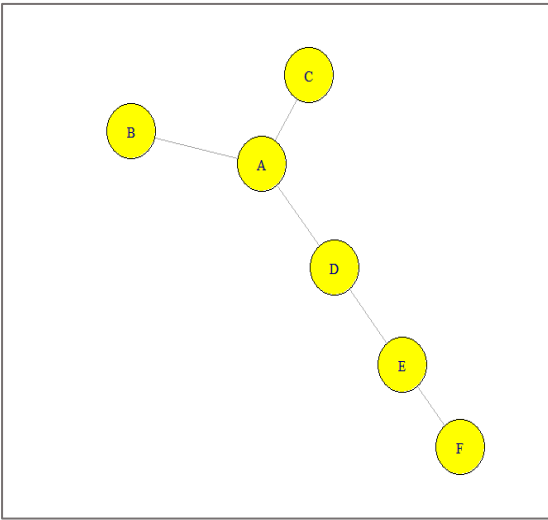
## Star형

	연결노드	연결노드수로 구하는 정도중심성
A	5	$5/(6-1)=1$
B	1	$1/(6-1)=0.2$
C	1	$1/(6-1)=0.2$
D	1	$1/(6-1)=0.2$
E	1	$1/(6-1)=0.2$
F	1	$1/(6-1)=0.2$



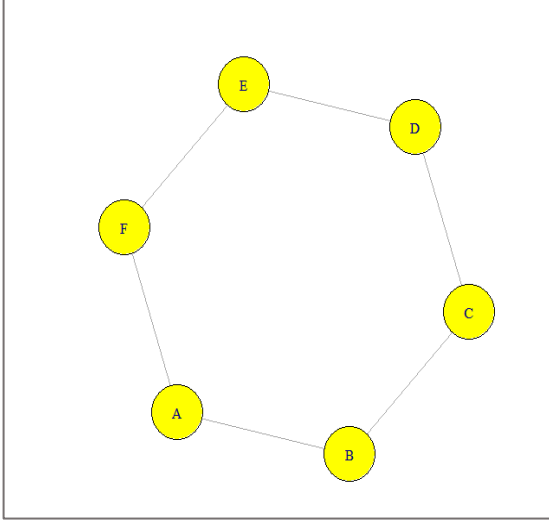
## Y형

	연결노드	연결노드수로 구하는 정도중심성
A	3	$3/(6-1)=0.6$
B	1	$1/(6-1)=0.2$
C	1	$1/(6-1)=0.2$
D	2	$2/(6-1)=0.4$
E	2	$2/(6-1)=0.4$
F	1	$1/(6-1)=0.2$



## 원형

	연결노드	연결노드수로 구하는 정도중심성
A	2	$2/(6-1)=0.4$
B	2	$2/(6-1)=0.4$
C	2	$2/(6-1)=0.4$
D	2	$2/(6-1)=0.4$
E	2	$2/(6-1)=0.4$
F	2	$2/(6-1)=0.4$



# 네트워크 구조별 연결정도 비교

[http://203.247.248.246/lectures/bigdata/\\_book/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC%EA%B7%B8%EB%9E%98%ED%94%84-%EC%9D%B4%EB%A1%A0.html#%EA%B7%B8%EB%9E%98%ED%94%84%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC%EC%9D%98-%EA%B5%AC%EC%84%B1](http://203.247.248.246/lectures/bigdata/_book/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC%EA%B7%B8%EB%9E%98%ED%94%84-%EC%9D%B4%EB%A1%A0.html#%EA%B7%B8%EB%9E%98%ED%94%84%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC%EC%9D%98-%EA%B5%AC%EC%84%B1)

#(n-1)(n-2) 이론적인 연결 정도 최대 중심화 계산  
centralization.degree.tmax(g\_star)  
centralization.degree.tmax(g\_Y)  
centralization.degree.tmax(g\_R)

#연결정도 중심화 측정  
centralization.degree(g\_star,normalized=FALSE)  
centralization.degree(g\_Y,normalized=FALSE)  
centralization.degree(g\_R,normalized=FALSE)

## Star형

	연 결 노 드	연결노드수로 구하 는 정도중심성	연결중심성 (max-각노드) Max=5
A	5	$5/(6-1)=1$	$5-5=0$
B	1	$1/(6-1)=0.2$	$5-1=4$
C	1	$1/(6-1)=0.2$	$5-1=4$
D	1	$1/(6-1)=0.2$	$5-1=4$
E	1	$1/(6-1)=0.2$	$5-1=4$
F	1	$1/(6-1)=0.2$	$5-1=4$
A: $0+4+4+4+4+4$			20
B: $(n-1)(n-2)$			$(6-1)(6-2)=20$
연결정도중심화 A/B			1

## Y형

	연 결 노 드	연결노드수로 구하는 정도중심성	연결중심성 (max-각노드) Max=3
A	3	$3/(6-1)=0.6$	$3-3=0$
B	1	$1/(6-1)=0.2$	$3-1=2$
C	1	$1/(6-1)=0.2$	$3-1=2$
D	2	$2/(6-1)=0.4$	$3-2=1$
E	2	$2/(6-1)=0.4$	$3-2=1$
F	1	$1/(6-1)=0.2$	$3-1=2$
A: $0+4+4+4+4+4$			8
B: $(n-1)(n-2)$			$(6-1)(6-2)=20$
연결정도중심화 A/B			0.4

## 원형

	연 결 노 드	연결노드수로 구하는 정도중심성	연결중심성 (max-각노드) Max=2
A	2	$2/(6-1)=0.4$	$2-2=0$
B	2	$2/(6-1)=0.4$	$2-2=0$
C	2	$2/(6-1)=0.4$	$2-2=0$
D	2	$2/(6-1)=0.4$	$2-2=0$
E	2	$2/(6-1)=0.4$	$2-2=0$
F	2	$2/(6-1)=0.4$	$2-2=0$
A: $0+4+4+4+4+4$			0
B: $(n-1)(n-2)$			$(6-1)(6-2)=20$
연결정도중심화 A/B			0

centralization.degree(g\_star,normalized=FALSE)\$centralization/centralization.degree.tmax(g\_star)  
centralization.degree(g\_Y,normalized=FALSE)\$centralization/centralization.degree.tmax(g\_Y)  
centralization.degree(g\_R,normalized=FALSE)\$centralization/centralization.degree.tmax(g\_R)

# igraph를 이용한 사회관계망 그래프(SNA: Social Network Analysis) 연습

[http://aispiration.com/network/ml-network-data.html#1\\_%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC\\_%EB%8D%B0%EC%9D%B4%ED%84%B0\\_%EA%B5%AC%EC%A1%B0](http://aispiration.com/network/ml-network-data.html#1_%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC_%EB%8D%B0%EC%9D%B4%ED%84%B0_%EA%B5%AC%EC%A1%B0)

<https://kuduz.tistory.com/1087>

[http://203.247.248.246/lectures/bigdata/\\_book/%EC%86%8C%EC%85%9C%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%ACsocial-network-%EC%97%BF%EB%B3%B4%EA%B8%B0.html](http://203.247.248.246/lectures/bigdata/_book/%EC%86%8C%EC%85%9C%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%ACsocial-network-%EC%97%BF%EB%B3%B4%EA%B8%B0.html)

## 네트워크 분석기법을 활용한 게임 데이터 분석

[https://danbi-ncsoft.github.io/works/2018/11/12/network\\_analysis-1.html](https://danbi-ncsoft.github.io/works/2018/11/12/network_analysis-1.html)

[https://danbi-ncsoft.github.io/works/2018/11/12/network\\_analysis-2.html](https://danbi-ncsoft.github.io/works/2018/11/12/network_analysis-2.html)

## 실습1: 문장간의 연관분석과 네트워크 그래프

```
test<-c("제주도 여행 제주도", "제주도 굴", "강사 고향 제주도", "강사 강아지 봄", "한국 설")
```

파일 편집 보기

제주도 여행 제주도  
제주도 굴  
강사 고향 제주도  
강사 강아지 봄  
한국 설

Txt 자료로 있는 경우에는  
readLines('txt파일명.txt')로 읽어야함.

```
splitList=strsplit(test,' ')
splitList
uniqueList=sapply(splitList, unique)
uniqueList
```

```
## 트랜잭션 데이터로 변환
tr=as(uniqueList, "transactions")
```

```
## 가로세로의 글자 cross테이블생성
crossTr=crossTable(tr)
crossTr
```

```
## 연관분석 실행 및 차트 생성
Rules <- apriori(crossTr, parameter=list(supp=0.1, conf=0.1))
Inspect(rules)
plot(rules,method="graph")
```

splitList  
1번째 제주도 중복

```
[[1]]
[1] "제주도" "여행"   "제주도"

[[2]]
[1] "제주도" "굴"

[[3]]
[1] "강사"   "고향"   "제주도"

[[4]]
[1] "강사"   "강아지" "봄"

[[5]]
[1] "한국"   "설"
```

unlist  
(splitList에서 중복제거)

```
[[1]]
[1] "제주도" "여행"

[[2]]
[1] "제주도" "굴"

[[3]]
[1] "강사"   "고향"   "제주도"

[[4]]
[1] "강사"   "강아지" "봄"

[[5]]
[1] "한국"   "설"
```

가로세로의 글자크로스 테이블 생성

```
> crossTr=crossTable(tr)
> crossTr
      강사 강아지 고향 굴 봄 설 여행 제주도 한국
강사      2      1      1      0      1      0      0      1      0
강아지      1      1      0      0      1      0      0      0      0
고향      1      0      1      0      0      0      0      1      0
굴          0      0      0      1      0      0      0      1      0
봄          1      1      0      0      1      0      0      0      0
설          0      0      0      0      0      1      0      0      1
여행      0      0      0      0      0      0      1      1      0
제주도      1      0      1      1      0      0      1      3      0
한국      0      0      0      0      0      1      0      0      1
>
```

## 실습1, 연관분석과 네트워크 그래프- 앞장에 이어서 (igraph를 이용한 SNA)

```
# (1) 데이터 구조 변경 : 연관규칙 결과 -> 행렬구조 변경(matrix 또는 data.frame)
rules.그래프 <- labels(rules, ruleSep=" ")
```

```
# 문자열로 묶인 연관단어를 행렬구조 변경
rules.그래프 <- sapply(rules.그래프, strsplit, " ", USE.NAMES=F)
```

```
# 행 단위로 묶어서 matrix로 반환
rulemat <- do.call("rbind", rules.그래프)
```

```
# (2) 연관어 시각화를 위한 igraph 패키지 설치
install.packages("igraph") # graph.edgelist(), plot.igraph(), closeness() 함수 제공
library(igraph)
```

```
# (3) edgelist보기 - 연관단어를 정점 형태의 목록 제공
ruleg <- graph.edgelist(rulemat[c(12:59),], directed=F) # [1,]~[11,] "{" 제외
ruleg
```

```
# (4) edgelist 시각화
plot.igraph(ruleg, vertex.label=V(ruleg)$name,
            vertex.label.cex=1.2, vertex.label.color='black',
            vertex.size=20, vertex.color='green', vertex.frame.color='blue')
```

