

# Social Network Analysis

단국대학교 산업공학과

**한고은**

goeun418@dankook.ac.kr



# CONTENTS

- 01 왜 Social Network Analysis이 필요한가?
- 02 Social Network Analysis 란?
- 03 Network 표현 방법
- 04 Social Network Analysis 지표
- 05 Social Network Analysis 예시

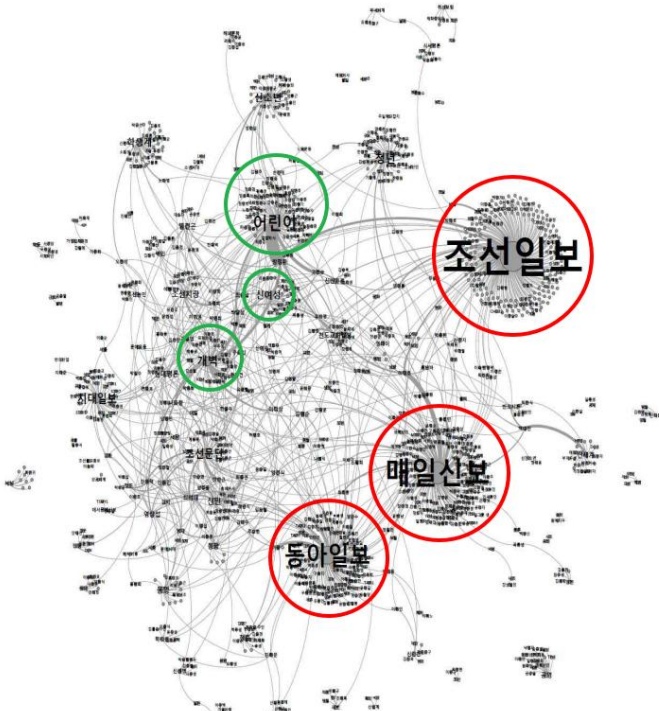
# 01 | 왜 Social Network Analysis이 필요한가?



# 01 | 왜 Social Network Analysis이 필요한가?



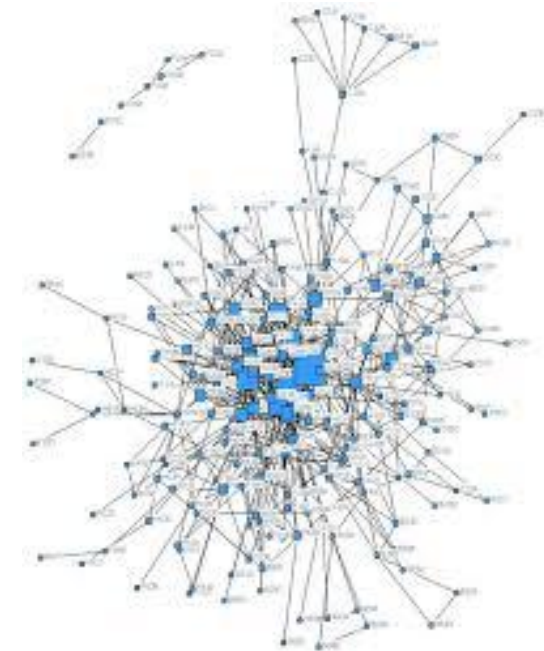
## • Network 예시



소설 출판 중심 작가-매체 연결망<sup>1)</sup>



페이스북 친구관계도<sup>2)</sup>



융합기술 특허 연결망<sup>3)</sup>

1) 매체 연결망 <https://wikidocs.net/214016>

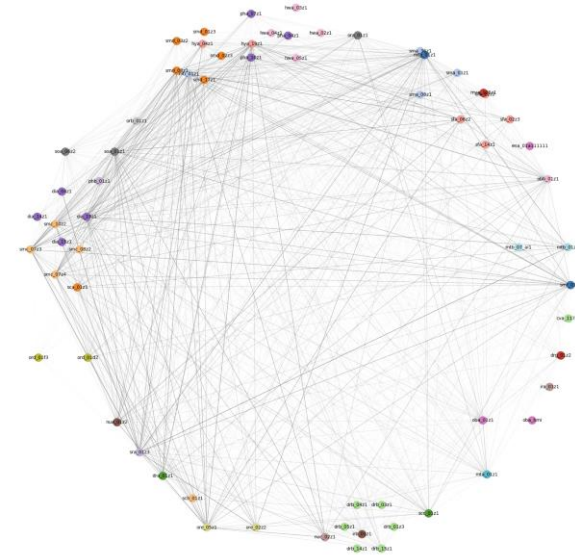
2) 페이스북 친구관계도 <http://www.theprconsulting.com/?p=4933>

3) 융합기술 특허 연결망 <https://www.jksqm.org/upload/pdf/jksqm-49-1-97.pdf>

## 02 | Social Network Analysis 란?



- **Social Network**
  - ✓ 개인, 집단, 사회의 관계
- **Social Network Analysis**
  - ✓ 사회 구성원 간의 관계 또는 상호작용하는 개체 간 관계의 패턴을 측정 및 분석



- **Social Network Analysis 관점**

- ✓ **구조적/관계적 접근법**

- 사회적 개체 간 관계의 구조적 형태와 내용을 파악 및 측정하여 가설을 검증
    - 기존 **개별적/변수중심적 분석방법**과 구별

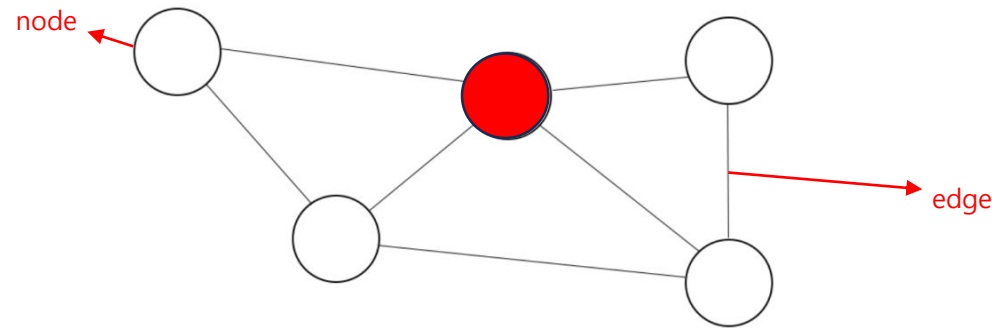
## 02 | Social Network Analysis 란?



- Network를 표현하기 위한 수학적 방법
  - ✓ Graph
  - ✓ Matrix

- Graph

- ✓ 두 개의 점이 선으로 연결되어 있는 형태



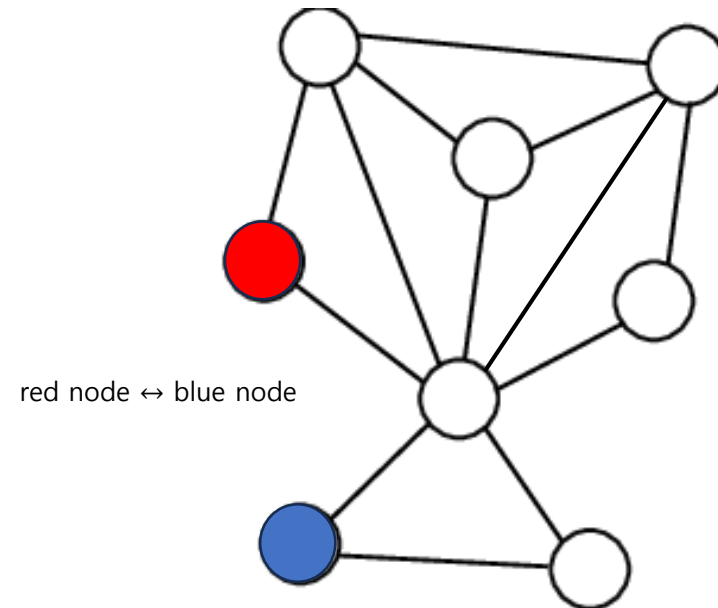
**Node** : graph에서 개별적인 개체(개인, 단체, 동물 등)

**Edge** : Node 사이에 존재하는 연결선

**Degree**: 특정 node에 연결된 모든 edge의 개수

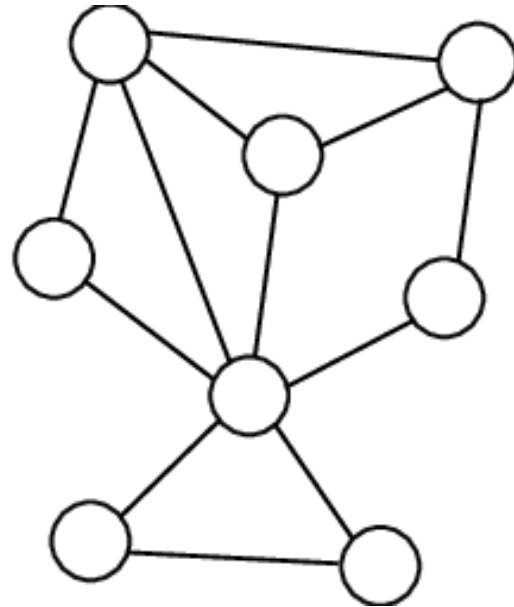
- Graph

- ✓ Edge의 유무

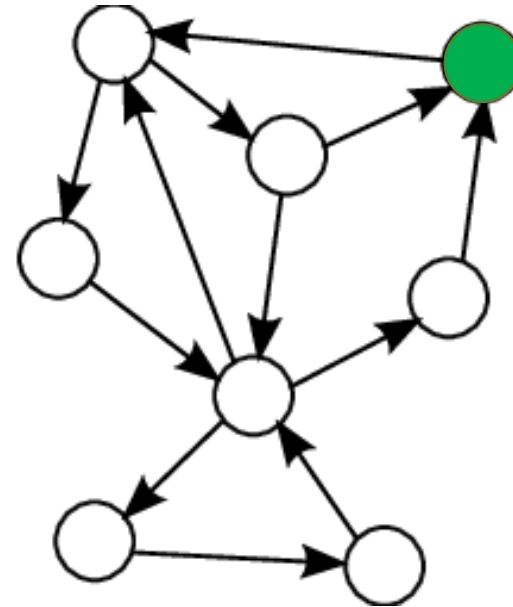


- Graph 종류

- ✓ 방향성



Undirected graph

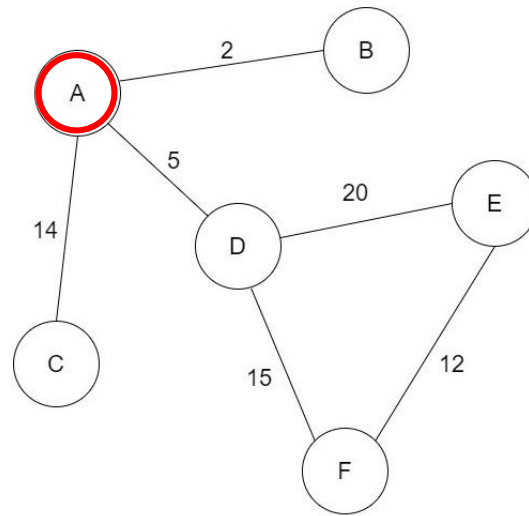


Directed Graph

In-degree: 들어오는 edge의 수  
Out-degree: 나가는 edge의 수

- Graph 종류

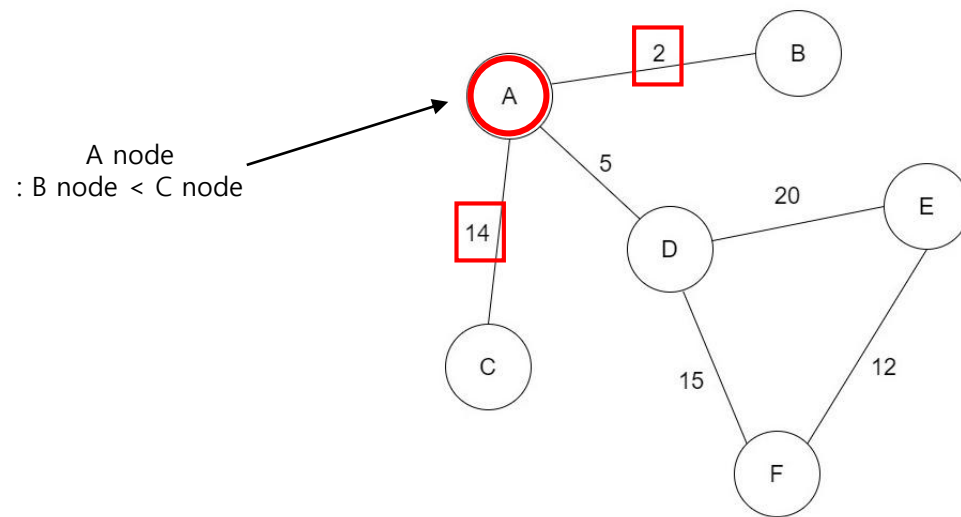
- ✓ 연결 정도



Weighted graph

- Graph 종류

- ✓ 연결 정도

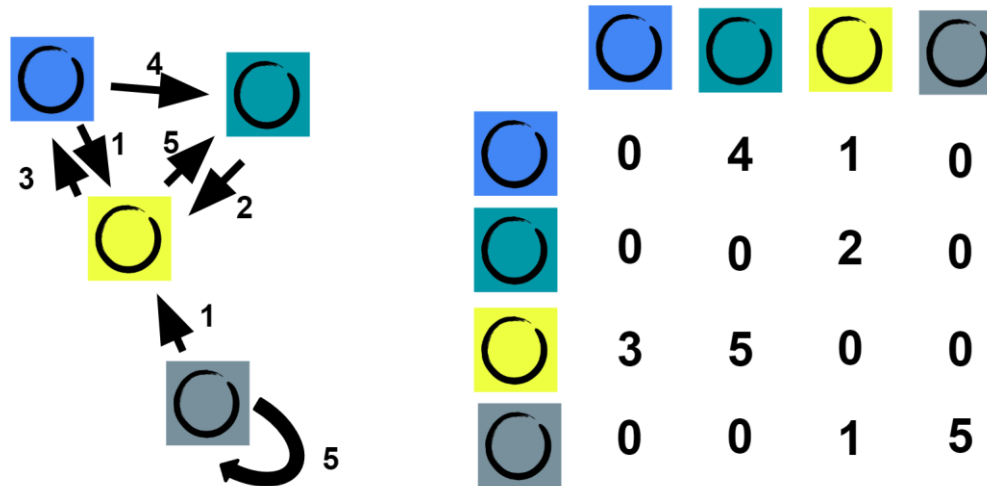


Weighted graph

- Matrix

- ✓ Adjacency matrix

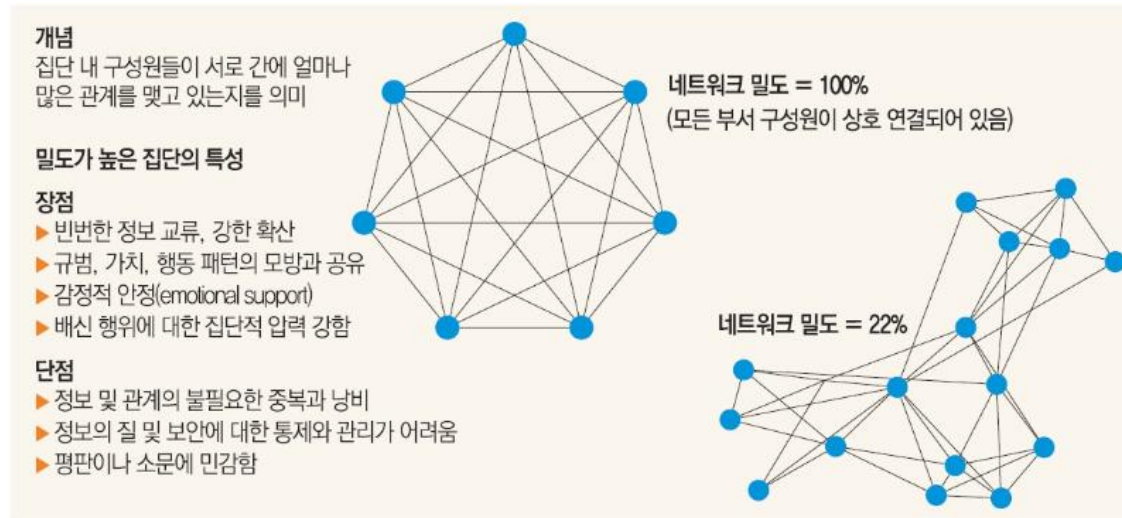
→ Node 간의 연결 관계를 행렬의 형태로 나타낸 것



- **분포(Distribution)**
  - ✓ 네트워크 내에서 다양한 노드들의 연결 상태
    - 밀도(Density), 중심성(Centrality) 등
- **세분화(Segmentation)**
  - ✓ 네트워크를 이루는 특정 그룹 분석
    - 클러스터링 계수(Clustering coefficient) 등

- 분포(Distribution)

- ✓ 밀도(Density): 네트워크에서 전체 노드들이 연결되어 있는 정도

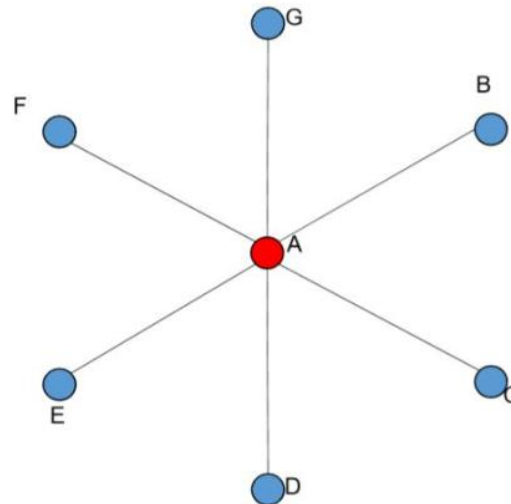


집단 내 구성원들의 네트워크<sup>4)</sup>

4) <https://cyram.tistory.com/6>

- 분포(Distribution)

- ✓ 중심성(Centrality): 네트워크에서 특정 노드가 얼마나 중요한지를 나타내는 방법



스타 네트워크

- 분포(Distribution)
  - ✓ 중심성(Centrality)
    - 연결 중심성(Degree Centrality)
    - 매개 중심성(Betweenness Centrality)
    - 근접 중심성(Closeness Centrality)
    - 아이겐벡터 중심성(Eigenvector Centrality)

- 중심성(Centrality)

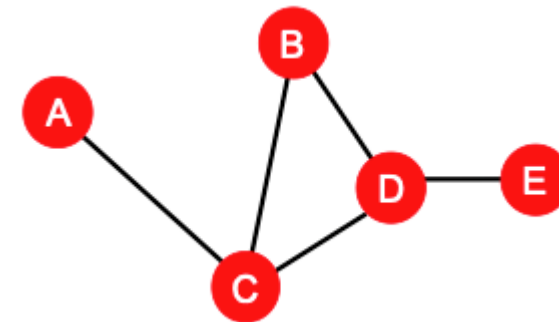
- ✓ 정도 중심성(Degree Centrality): 네트워크에서 다른 노드에 연결되어 있는 정도를 측정하는 방법

$$C_d(v) = \frac{d_v}{|N| - 1} = \frac{v\text{의 } degree}{\text{전체 노드 개수} - 1}$$

- 중심성(Centrality)

- ✓ 매개 중심성(Betweenness Centrality): 두 노드 간의 최단 경로 위에 해당 노드가 얼마나 자주 있는지를 표현하는 방법

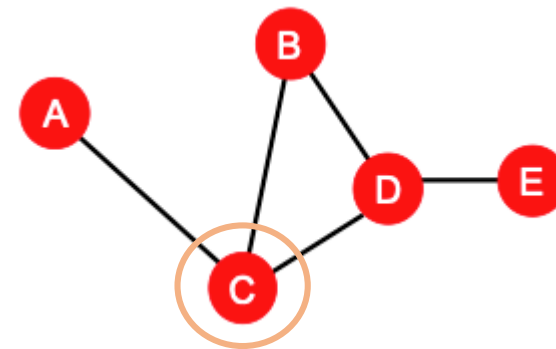
$$C_b(v) = \sum_{s,t \in N} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$
$$= \sum_{s,t \in N} \frac{\text{노드 } s \rightarrow t \text{ path 중, } v \text{ 를 거쳐가는 path 개수}}{\text{노드 } s \rightarrow t \text{ path 개수}}$$



- 중심성(Centrality)

- ✓ 매개 중심성(Betweenness Centrality): 두 노드 간의 최단 경로 위에 해당 노드가 얼마나 자주 있는지를 표현하는 방법

$$C_b(v) = \sum_{s,t \in N} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$
$$= \sum_{s,t \in N} \frac{\text{노드 } s \rightarrow t \text{ path 중, } v \text{ 를 거쳐가는 path 개수}}{\text{노드 } s \rightarrow t \text{ path 개수}}$$

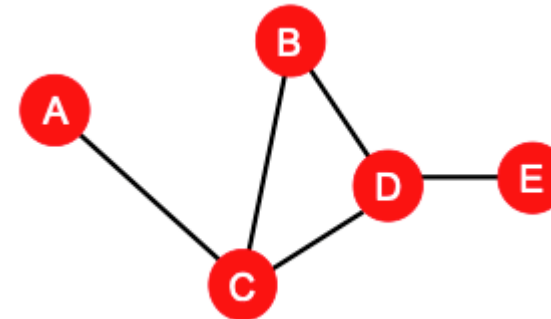


- 중심성(Centrality)

- ✓ 근접 중심성(Closeness Centrality): 특정 노드와 모든 노드들 사이의 최단 경로의 길이를 표현하는 방법

$$C_c = \frac{|N| - 1}{\sum_{u \in N \setminus \{v\}} d(v, u)}$$

=  $\frac{\text{전체 노드 개수} - 1}{v \text{에서 다른 노드들까지의 최단 경로의 총합}}$

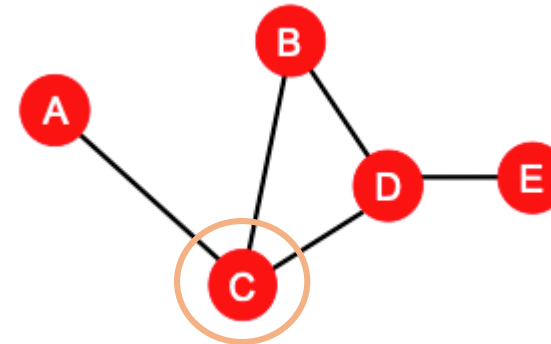


- 중심성(Centrality)

- ✓ 근접 중심성(Closeness Centrality): 특정 노드와 모든 노드들 사이의 최단 경로의 길이를 표현하는 방법

$$C_c = \frac{|N| - 1}{\sum_{u \in N \setminus \{v\}} d(v, u)}$$

=  $\frac{\text{전체 노드 개수} - 1}{v\text{에서 다른 노드들까지의 최단 경로의 총합}}$



- 중심성(Centrality)

- ✓ 아이겐벡터 중심성(Eigenvector Centrality): 이웃한 노드들의 중요도를 표현하는 방법

$$C_E(v) = \frac{1}{\lambda} \sum_j A_{ij} C_E(v_i)$$

$C_E(v_i)$       노드  $v_i$ 의 eigenvector centrality 값

$A_{ij}$       인접 행렬의 값

$\lambda$       행렬  $A$ 의 eigenvalue

# 04 | Social Network Analysis 지표

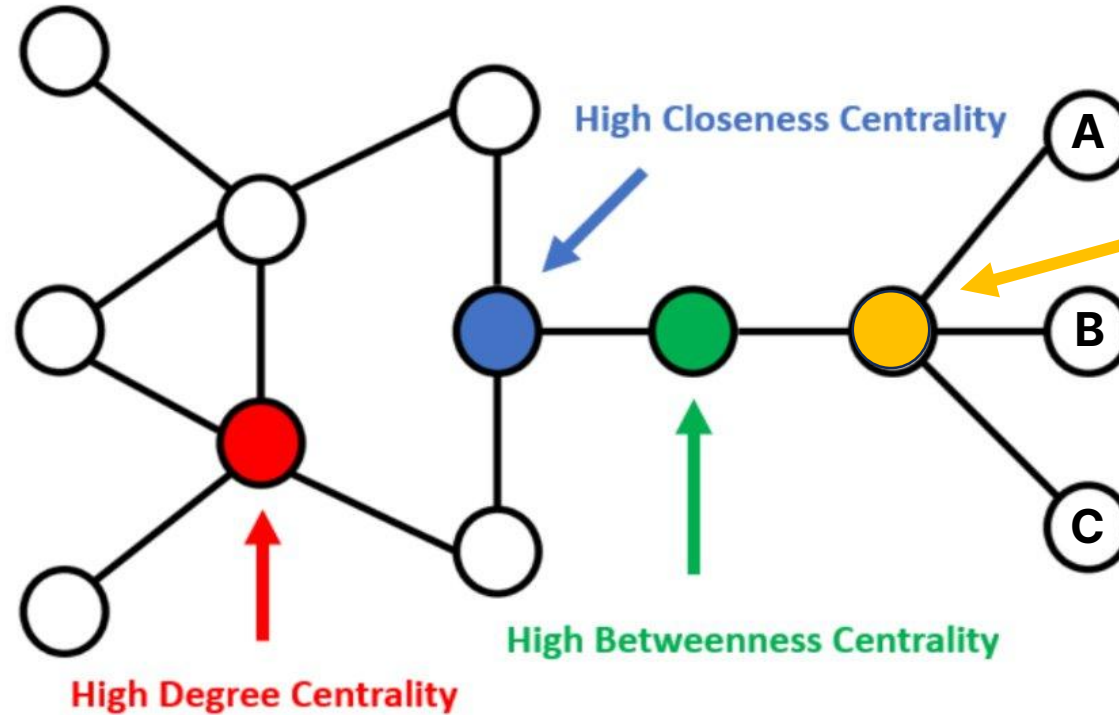


## • 중심성(Centrality)

✓ 아이겐벡터  
는 방법

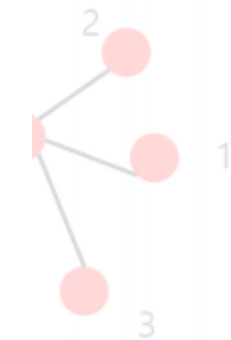
$C_E(v)$

$C_E(v_i)$ : 노드  $v_i$ 의  
 $A_{ij}$ : 인접 행렬의  
(노드  $v_i$ 와  $v_j$ 가)  
 $\lambda$ : 특정 eigenvalue



요도를 표현하

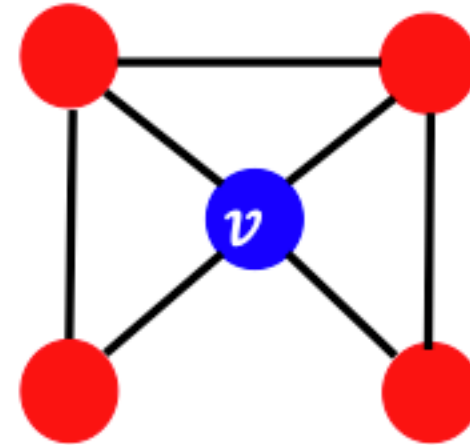
High Eigenvector Centrality



- 세분화(Segmentation)

- ✓ 클러스터링 계수(Clustering coefficient): 네트워크 내에서 특정 노드의 클러스터링 정도를 측정

$$c_u = \frac{|(v_1, v_2) \in \mathcal{E} : v_1, v_2 \in \mathcal{N}(u)|}{\binom{d_u}{2}}.$$



## 05 | Social Network Analysis 예시



- 8명의 응답자(A,B,C,D,E,F,G,H)
- 3개의 질문
- 각 질문에 동일한 응답을 한 사람들 간에 edge를 생성

Question 1	Do you exercise regularly?
Question 2	Do you maintain a healthy diet?
Question 3	Do you get enough sleep?

	Question 1	Question 2	Question 3
A	Yes	No	YES
B	No	YES	YES
C	YES	No	No
D	YES	No	YES
E	No	YES	No
F	YES	YES	YES
G	No	YES	No
H	YES	No	YES

## 05 | Social Network Analysis 예시



- 8명의 응답자(A,B,C,D,E,F,G,H)
- 3개의 질문
- 각 질문에 동일한 응답을 한 사람들 간에 edge를 생성

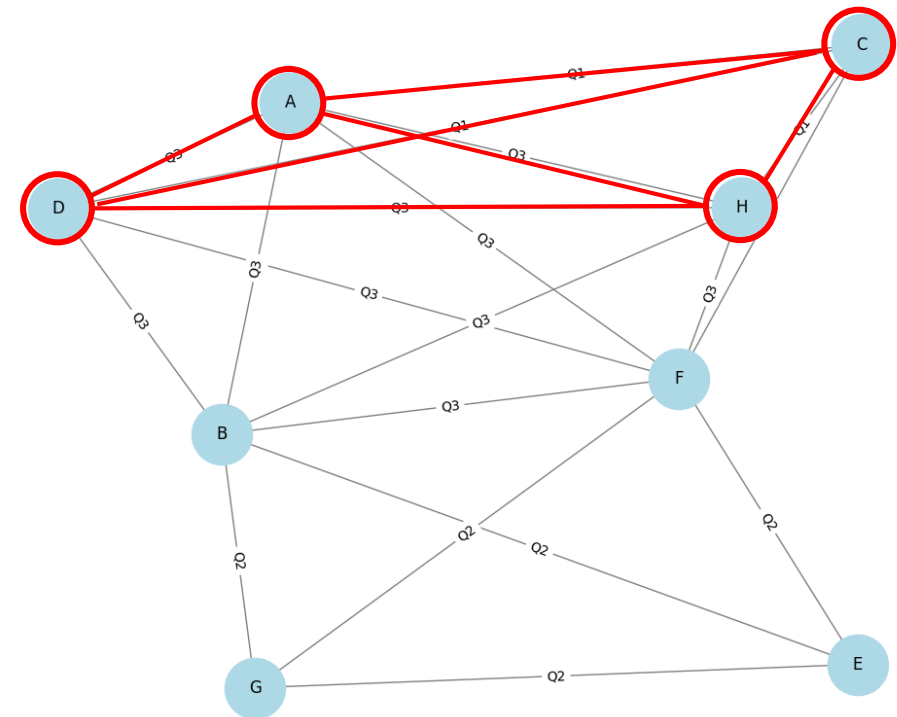
```
# 각 질문에 대해 'Yes'라고 응답한 사람들 간의 엣지를 추가
for question in ['Q1', 'Q2', 'Q3']:
    # 'Yes'라고 응답한 사람들의 리스트 생성
    yes_respondents = [person for person, answers in responses.items() if answers[question] == 'Yes']

    # 'Yes'라고 응답한 사람들끼리 엣지 추가
    for i in range(len(yes_respondents)):
        for j in range(i + 1, len(yes_respondents)):
            person1 = yes_respondents[i]
            person2 = yes_respondents[j]
            edges.append((person1, person2, question))
```

Question 1	Do you exercise regularly?
Question 2	Do you maintain a healthy diet?
Question 3	Do you get enough sleep?

	Question 1	Question 2	Question 3
A	Yes	No	YES
B	No	YES	YES
C	YES	No	No
D	YES	No	YES
E	No	YES	No
F	YES	YES	YES
G	No	YES	No
H	YES	No	YES

SNA: Respondents with 'Yes' Answers by Question and Centrality

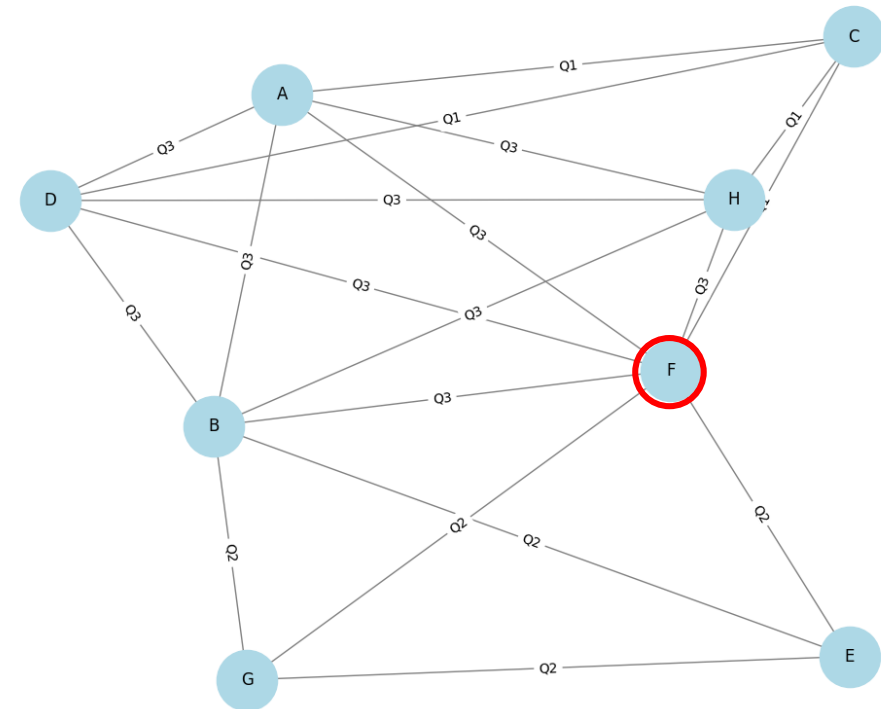


**Degree centrality****A: 0.714****B: 0.857****C: 0.571****D: 0.714****E: 0.429****F: 1.000****G: 0.429****H: 0.714**

```
# 중심성 계산
degree centrality = nx.degree_centrality(G)

# 중심성 값 출력
print("Degree Centrality:")
for node, centrality in degree_centrality.items():
    print(f"{node}: {centrality:.3f}")
```

SNA: Respondents with 'Yes' Answers by Question and Centrality



**감사합니다.**