서울대학교 인지과학전공
네트워크 사이언스 실습

강사 최동혁

최동혁

디지털 인문학 연구자 | 개발자 | 네트워크 과학자

사학 / 컴퓨터공학 전공 (2016)
KAIST 문화기술대학원 석사 (2018)
KAIST 문화기술대학원 박사 (2024)

연구주제: Digital Humanities, Digital history, Quantitative history, Social network, Quantifying success, 조선시대 사회사, 정치사, 사상사

그 밖의 관심사: 영화, 배낭여행, 음주, 코딩, 뉴스, 나무위키 서핑

네트워크 사이언스는?

Networks: An Introduction, Mark Newman (2018, 2nd Edition) 목차

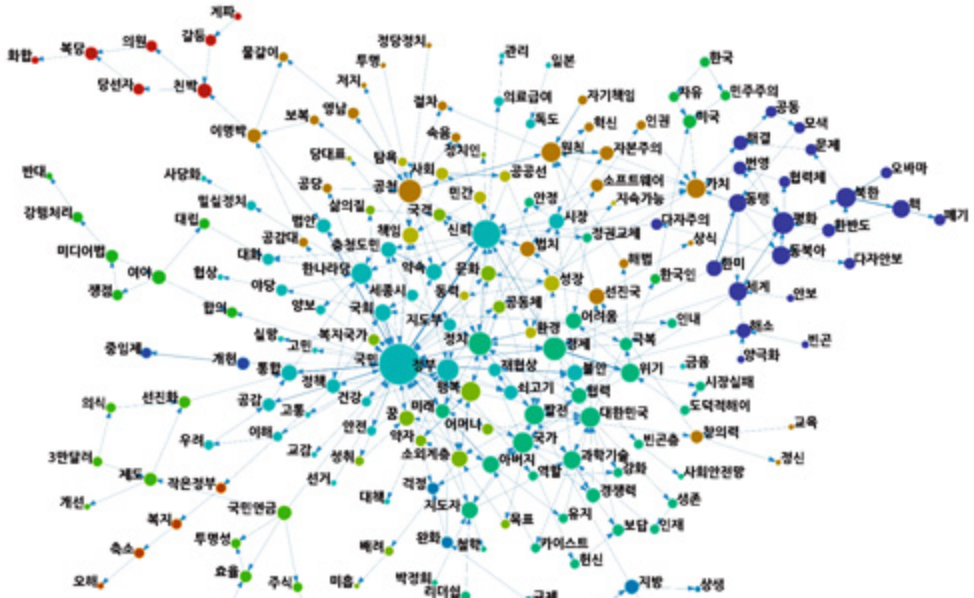
네트워크 사이언스에 대한 일반적 인식

출처: 네트워크 과학이 밝힌 박근혜 블랙박스

네트워크 과학이 밝힌 박근혜 블랙박스

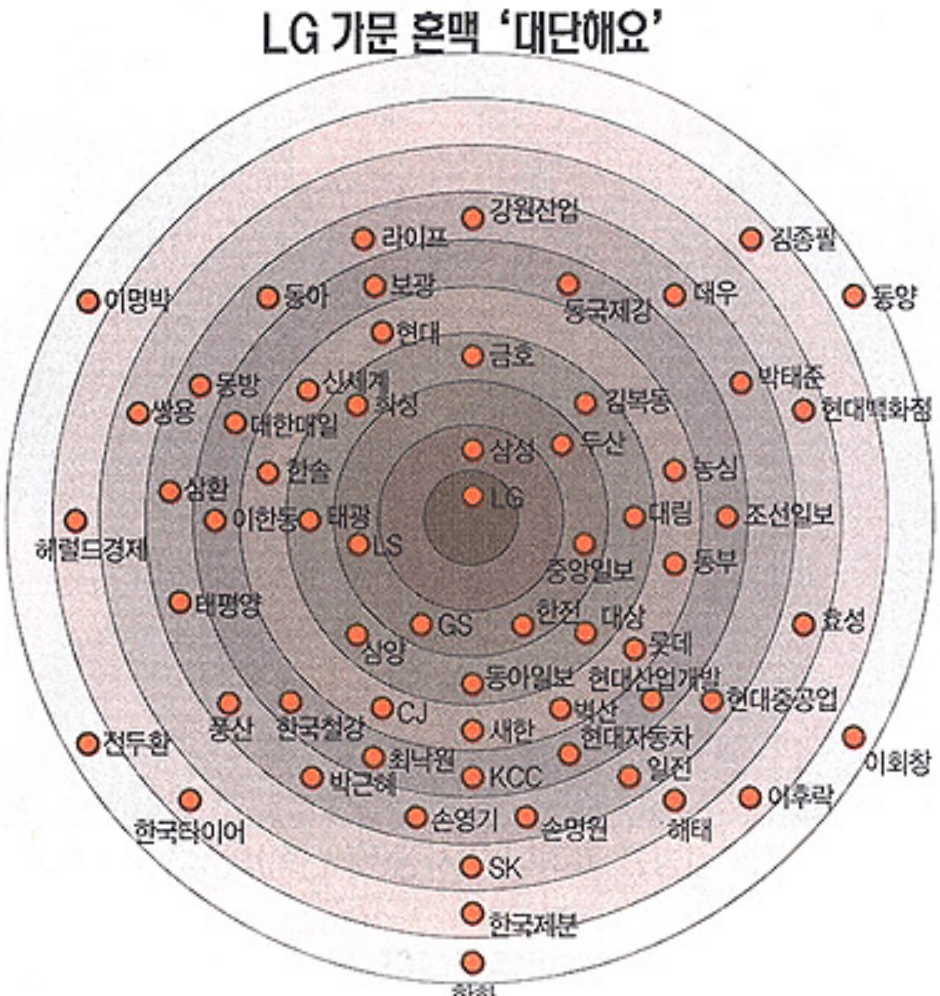
4·27 재·보선 이후 박근혜 전 대표의 일거수일투족에 관심이 쏠리고 있다. 그런데 과연 정치인 박근혜는 제대로 평가되고 있는 것일까.

천관율 기자 다른기사 보기 > | 입력 2011.05.16 09:40 수정 2021.11.17 16:31 191호



네트워크 사이언스에 대한 일반적 인식

출처: 재벌 혼맥의 LG가 '허브'



네트워크 사이언스에 대한 일반적 인식

- 데이터만 있으면 네트워크 과학으로 돌리면 새로운 사실이 드러난다?
- 서로 모를 것 같았던 사람들이 알고 보니 연결되어 있었다?

오해의 원인

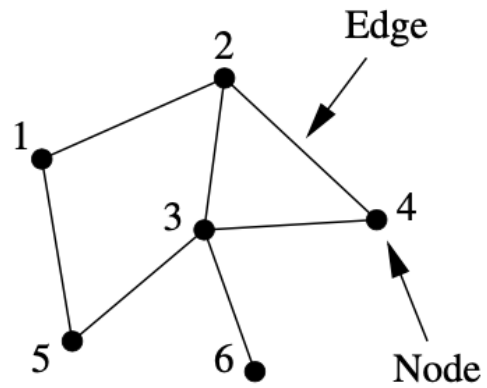
- 이른바 '빅 데이터 (Big Data)'가 불러온 환상
  - *"데이터 넣고 돌리면 알아서 결과가 나온다"*
  - 데이터 분석이 복잡한 수학적 이론을 기반으로 하기 때문에 '마법'으로 여겨짐.
- 네트워크 사이언스 이론의 대중성 부족
  - 네트워크 사이언스는 오래되었지만, 1990년대 이후로 급격히 발전하여 대중성이 낮음.
  - 앞의 사례들은 "Small World Phenomenon"으로 설명 가능.

네트워크 기초용어(Network Basics)

점과 선

용어	네트워크 사이언스	수학	사회학	물리학
점	nodes	vertices	actors	sites
선	edges	links	ties	bonds

노드 연결 행렬 (Adjacency Matrix)



$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

- 왼쪽 네트워크를 행렬로 표현하면 오른쪽 A와 같이 표현할 수 있다.
- 즉, 행과 열을 노드의 번호라고 할 때, 노드끼리 엣지가 있으면 1, 없으면 0으로 표현한다.
- 대각선 원소들은 0으로 처리하는데, 만일 자기 자신과 연결되는 노드(self-node)라면, 1로 처리해도 무방하다.

노드 연결 리스트(Adjacency List)

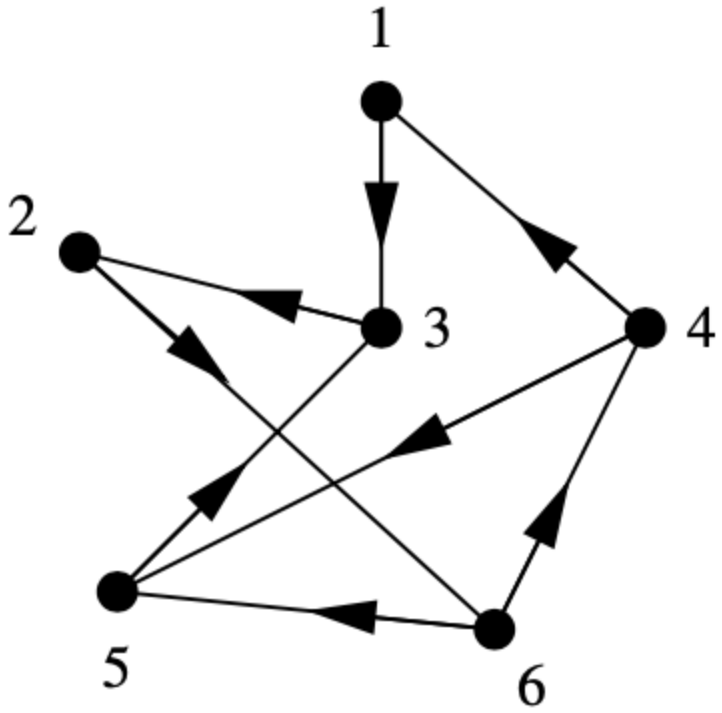
- 노드 연결 행렬로 표현하면, 직관적으로 이해하기 쉽다는 장점이 있지만, 일상의 대부분의 네트워크들은 연결이 많지 않은 희소 행렬(Sparse matrix)인 경우가 많다.
- 행렬의 크기가 하나씩 커질수록 컴퓨터는 $O(n^2)$ 즉, 제곱만큼의 연산을 더 해야 한다. 이러한 비효율을 줄이기 위해 고안된 것인 노드 연결 리스트이다.

```
1: [2, 5]
2: [1, 3, 4]
3: [2, 4, 5, 6]
4: [2, 3]
5: [1, 3]
6: [3]
```

또는 Edge List라고 하여 다음과 같이 표현할 수 있다.

```
[1, 2][(1, 5)][(2, 3)][(2, 4)][(3, 4)][(3, 5)][(3, 6)];
```

방향이 있는 네트워크(Directed network)



- 노드 A가 다른 노드 B를 가리키는 네트워크는 방향이 있다고 하며 이렇게 표현되는 네트워크를 **방향이 있는 네트워크(Directed network)** 라고 부른다.
- 이 네트워크에서는 양쪽 노드가 서로 가리켜서 양방향(Bi-directional) 관계가 만들어 질 수도 있다.
- 방향이 있는 네트워크와 방향이 없는 네트워크에 적용되는 알고리즘과 방법론이 상당히 차이가 있으니 반드시 주의가 필요하다.

네트워크 측정 지표(Network Measures and Metrics)

차수(Degree)

1. **차수(degree)**

   $k_i = \sum_{j=1}^{n} A_{ij}$

   정의: 노드에 연결된 엣지의 수

   (가중치가 없는 경우) **노드 연결 행렬**에서 나의 열(또는 행)에 해당하는 원소의 합

   (참고) 허브(Hub): 가장 차수가 높은 노드

2. **평균 차수(average degree)**

$$c = \frac{1}{n}\sum_{i=1}^{n} k_i$$

정의: 모든 노드 차수의 평균

(가중치가 없는 경우) 노드 연결행렬에서 모든 원소의 합을 노드의 숫자만큼 나눈 것

3. **들어오는 차수(in-degree)**

정의: 방향이 있는 네트워크에서 노드로 들어오는 엣지의 수

4. **나가는 차수(out-degree)**

정의: 방향이 있는 네트워크에서 노드에서 나가는 엣지의 수

5. **밀도(Density)**

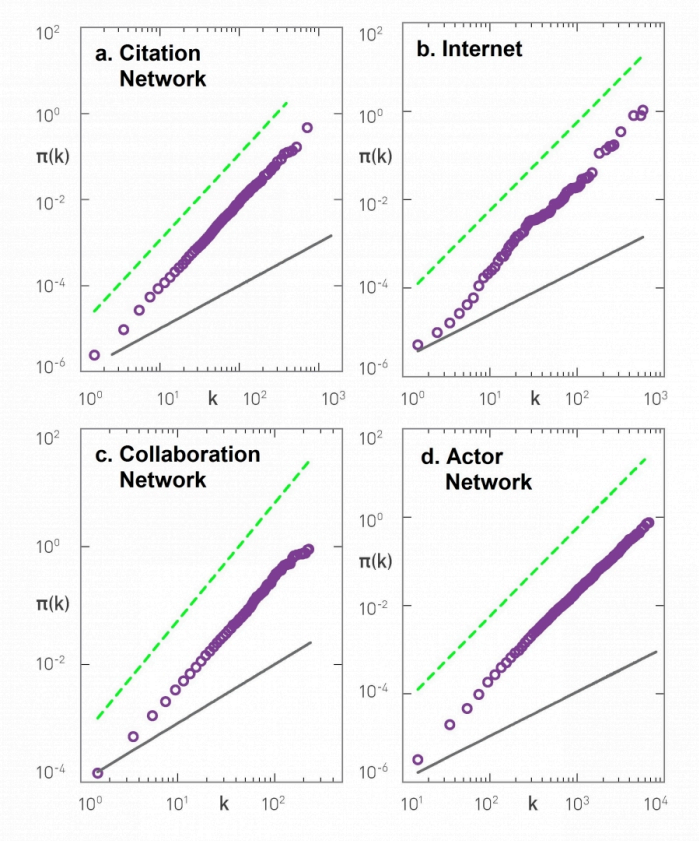
$$\rho = \frac{2m}{n(n-1)} = \frac{c}{n-1}$$

여기서 m은 전체 엣지의 수

정의: 최대 가능한 엣지의 수와 실제 엣지의 수의 비율

6. 차수 분포(Degree Distribution)

- 차수 분포는 왠지 정규분포를 따를 것 같지만, 실제 대부분의 네트워크들은 차수가 많은 노드는 굉장히 적고, 차수가 적은 노드들은 굉장히 많은 형태의 분포를 갖는다. 이런 네트워크를 Scale-free network 라고 부른다.



경로(Path)

- 네트워크에서 노드 A와 B가 있다고 할 때, A에서 B로 가는 엣지의 집합을 경로라고 부른다. 경로는 여러 개가 있을 수 있다.

최단 경로(Shortest path)

- 경로 중 가장 값이 작은 것을 최단 경로라고 한다.

평균 경로 길이(Average shortest path)

- 모든 최단 경로의 평균을 평균 경로 길이라고 한다.

지름(Diameter)

- 가장 긴 최단 경로를 diameter라고 한다.

중심성(centrality)

- 중심성은 네트워크에서 가장 중요한 노드가 무엇인지 그 노드의 속성과 무관하게 네트워크의 위상학(network topology)적 특성만으로 찾아내는 방법이다.

1. **차수 중심성(Degree centrality)**

> 차수가 높을수록 중요하다. (마당발)

2. **고유 벡터 중심성(Eigenvector centrality)**

> 차수가 많은 노드들과 많이 연결되어 있을수록 중요하다. (흑막?)

$$x_i = \kappa^{-1} \sum_{i\text{의 이웃노드} j} x_j$$

나와 연결된 노드들의 점수들을 계산해야 내 노드를 구할 수 있다면, 가장 처음 점수는 어떻게 구하는가? ⇒ eigenvector와 관련된 개념들. 모르면 넘어가도 괜찮다.

3. 끼임 중심성(Betweenness centrality)

> 경로에 많이 놓여 있을수록 중요하다. (브로커)

$$x_i = \sum_{st} \frac{n_{st}^i}{g_{st}}$$

g_{st}: 노드 s에서 t까지 가는 최단 경로 길이의 총합(즉, 모든 네트워크 최단 경로 길이의 총합)

n_{st}^i: 노드 i를 거쳐가는 노드 s에서 t까지 가는 최단 경로의 총합

4. 근접 중심성(Closeness centrality)

> 다른 모든 노드들과의 최단 경로가 짧을수록 중요하다. (인싸)

$l_i = \frac{1}{n}\sum_j d_{ij}$ (평균 최단 경로)

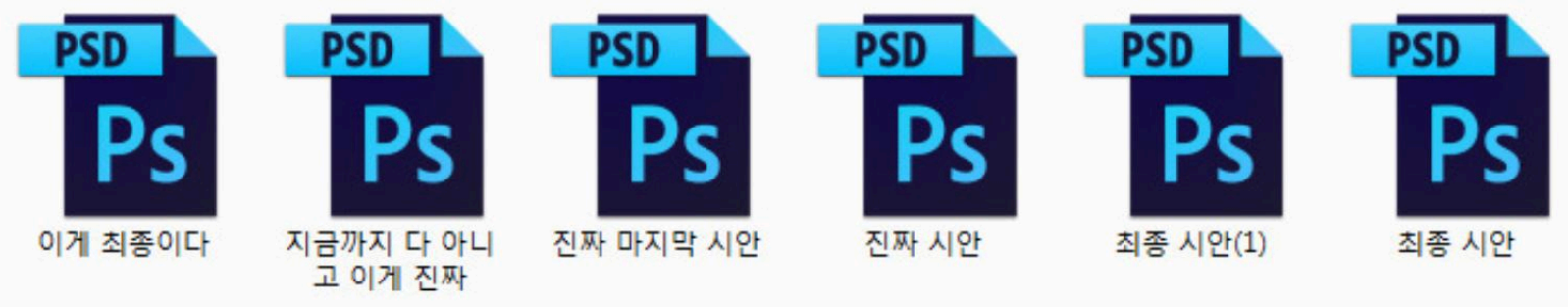
d_{ij}: 나(i)에서 다른 노드(j)까지의 최단경로의 합

$C_i = \frac{1}{l_i} = \frac{n}{\sum_j d_{ij}}$

l_i의 역수가 근접 중심성이다.

1. Git과 Github

Git은 왜 필요할까?



- 우리는 일을 하면서 수많은 선택의 기로에 놓인다.
- (일반적인 해결 방법) 새로운 "버전"을 만들어서 파일이름에 그 버전의 특징을 기록하기
- (Git을 이용한 해결 방법) 버전을 만들어서 저장해두고, 필요할 때마다 그 버전으로 돌아간다.
- (Git을 이용한 협업 방법) 여러 사람의 작업을 일종의 '버전'으로 생각하고 그 버전을 합치는 방법을 제공한다.

Git과 Github의 차이는?

- Git은 **내 컴퓨터 안(Local)에서 버전관리를 도와주는 프로그램**
- Github은 Git을 조금 **더 편리하게 사용할 수 있게 도와주는 플랫폼(Remote, Server)**
  - 여러 사람들이 함께 Git을 사용할 수 있다.
  - 나의 Git을 백업할 수 있다. (심지어 북극에도!)
- Git이 동영상이라면, Github은 유튜브라고 생각하면 된다.

실습: 윈도우 환경에서 Git 설치하기

실습: 윈도우 환경에서 Github Desktop 설치하기

2. VS Code

- VS Code는 Microsoft에서 만든 오픈소스 텍스트 에디터
- 텍스트 에디터이지만, 플러그인을 설치하면 IDE처럼 사용할 수 있다.
- 전세계에서 개발자들이 가장 인기 있는 텍스트 에디터이다.

(참고) 사티아 나델라, 그가 박수 받을 수 있었던 이유...MS 퓨처나우

실습: 윈도우 환경에서 VS Code 설치하기

실습: VS Code에서 Extension 설치하기

실습: VS Code에서 Jupyter notebook 실행하기

3. Gephi

- Gephi는 가장 인기 있는 네트워크 시각화 도구
- Java가 설치되어 있어야 한다.
- 발음에 유의하자.
  - 제피, 제파이 지피, 게파이가 아니다.
  - (참고1)
  - Gephi 공식계정의 입장
  - Gephi 발음
  - (참고2) Latex의 발음은 '라텍스'가 아님 주의

데이터 소개

- 열린국회정보 API를 이용한 제21대 국회의원 네트워크 분석

열린국회정보 API

- 국회에서 제공하는 API로 국회의원, 의안, 회의록 등 총 192건의 API를 제공한다.
- API 사용신청
- 실습에 활용할 API는 1. 국회의원 발의 법률안, 2. 국회의원 인적사항 이다.

용어 정리

모듈: 자소서 한 문단

- 모듈은 재사용 가능한 코드 집합
- 마치 자소서 한 문단이 특정 주제를 다루는 것처럼, 모듈은 특정 기능이나 요소를 담고 있습니다.
  예시: math, time, random, 내가 짠 모듈, 자소서의 한 문단

패키지: 자소서 한 장

- 패키지는 여러 모듈을 포함하는 컨테이너입니다.
- 자소서에 비유하면, 자소서는 여러 문단(모듈)이 모여서 만들어 졌다고 볼 수 있습니다.
  예시: openai, ...

라이브러리: 자소서 폴더

- 라이브러리는 특정 작업을 위한 함수, 클래스, 모듈의 집합입니다. 도서관처럼 다양한 자료(모듈, 패키지)를 갖추고 있어 필요할 때마다 가져다 사용할 수 있다.
- 라이브러리와 패키지는 종종 혼용되어 사용되지만, 일반적으로 라이브러리는 여러 모듈과 패키지의 집합체로 이해된다. 반면 패키지는 하나의 단위로 배포되고 관리되는 모듈의 그룹이다.
- 자소서 폴더 안에는 자소서 한 문단도 있을 수 있고, 한 장의 자소서도 있을 수 있다.
  예시: PyTorch, Pandas, Numpy, Scikit-learn, matplotlib

프레임워크: 선배가 준 자소서 폴더

- 프레임워크는 기본적인 구조와 가이드 문서를 제공하는 프로그래밍 환경이다.
- 자소서에 비유하면, 이미 취업에 성공한 선배가 만들어 놓은 자소서가 있어서 우리는 그 자소서에서 내 상황에 맞는 내용만 갈아 끼우면 되는 상황을 생각해 보면 된다.
- '제어역전'의 개념이 포함되어, 사용자가 작성하는 코드는 프레임워크에 의해 호출된다.
- **예시**: Django, Flask, ...

SDK: 자소서 만들기 어플

- SDK(Software Development Kit)는 특정 소프트웨어, 플랫폼, 프레임워크를 개발할 때 필요한 도구와 라이브러리의 집합이다.
- 자소서에 비유하면, SDK는 자소서 작성을 위해 필요한 모든 템플릿, 자주 쓰는 내용들, 꿀팁 등을 제공해 주는 어플을 생각해 볼 수 있다.
  예시: Java SDK, Philips Hue SDK, ...

API: 자소서 문단의 이름에 대한 설명 문서

- API(Application Programming Interface)는 소프트웨어 간 상호작용을 위한 인터페이스이다.
- 자소서에 비유하면, “성장배경”, “학력” 등의 문단들이 있을 때, 각 문단의 이름에 대한 설명이 있는 문서를 생각해 볼 수 있다.

  예시: REST API, OpenAI API, Google Map API, ...

데이터 수집

웹 크롤링(Web Crawling)이란?

- **웹 크롤링**은 웹 페이지를 **자동**으로 요청하여 원하는 정보만 취득하는 기술
- 크롤러(Crawler)라가 웹 페이지를 순회하며 데이터를 수집하고 저장한다.
- 구글은 전세계에서 가장 열심히 웹 크롤링을 하는 회사 중 하나이다!

웹 크롤링으로 뭘 할 수 있나?

- 뉴스 수집
- 쇼핑몰 상품 가격 추적
- 강의 계획서 모으기 등등

꼭 지켜줘요.

- 일부 웹 사이트는 크롤링을 금지하거나 사용량에 제한을 두고 있습니다. robots.txt 파일을 꼭 확인하세요.
- 구글의 크롤링 정책 보기
- 네이버의 크롤링 정책 보기
- 아마존의 크롤링 정책 보기
- 쿠팡의 크롤링 정책 보기

실습: 국회의원 발의 법률안 데이터, 국회의원 인적사항 데이터 수집하기 (0_crawl.ipynb)

실습: 데이터 전처리 하기 (1_preprocessing.ipynb)

데이터 분석 (기초)

가설검정

- H_0: 귀무가설(null hypothesis). 처음부터 버릴 것을 예상하는 가설. 샘플들 사이에 차이가 없다는 가설이다.
- H_a: 대립가설(alternative hypothesis). 샘플들 사이에 통계적으로 유의한 차이가 있다는 가설이다.
- (예시) 귀무가설: 대한민국 남자와 여자의 키는 같다. 대립가설: 대한민국 남자의 키가 여자의 키보다 더 크다.
- (해설) 만일 대한민국 남자와 여자의 키가 다르다는 주장을 하고 싶다면, 우선 귀무가설을 세우고, 그 귀무가설을 기각할 증거를 찾아야 한다.

1. 양대 정당의 발의 법률안의 수가 같을까?

- H_0 : 양대 정당은 발의한 법률안의 수가 같다.
- H_a : 양대 정당은 발의한 법률안의 수가 다르다.

2. 다선의원일수록 법률안을 많이 발의할까?

- H_0 : 국회의원은 당선된 횟수와 관계없이 발의한 법률안의 수가 같다.
- H_a : 국회의원의 당선된 횟수는 발의한 법률안의 수와 관계가 있다.

실습: 데이터 분석하기 (2_basic_analysis.ipynb)

네트워크 분석

3. 여성 의원은 남성 의원보다 중재자 역할을 더 많이 할까?

- H_0: 성별에 따라 중재자 역할의 차이는 없다.
- H_a: 성별에 따라 중재자 역할의 차이가 있다.

4. 제20대 국회와 비교하여 제21대 국회가 더 많이 응집되었을까?

- H_0: 제20대 국회와 제21대 국회의 뭉침계수는 같다.
- H_a: 제20대 국회와 제21대 국회의 뭉침계수는 다르다.

실습: Gephi로 네트워크 시각화하기 (Gephi에서 진행, 3_lawmaker.gephi)

실습: 네트워크 분석하기 (4_network_analysis.ipynb)

결론

- 네트워크 사이언스는 결국 데이터 분석 방법론의 일부이다.
- 어떤 연구질문(가설)을 가졌는지에 따라 데이터 분석의 방법은 달라질 수 있고, 그래야만 한다.
- 네트워크 사이언스를 활용하여 데이터에서 찾을 수 있는 특성으로는 1. Complexity, 2. Emergent property, 3. Centrality 등이 있다.