빅데이터전문가

한병준 교수

KIET 한국교육검정원
KOREA INSTITUTE FOR EDUCATION AND TEST

1강. 빅데이터의 개념

1. 빅데이터란 무엇인가?

● 빅데이터 (Big Data)

빅데이터는 기존의 데이터 처리 응용 소프트웨어가 처리하기에 매우 거대(large)하거나 복잡한(complex) 데이터 집합을 뜻한다.

과거 데이터를 취급하기 위해서는 데이터베이스 관리 시스템(DBMS)이나 데이터 웨어하우스(data warehouse)의 구축이 중요하였다. 그러나 시대가 흐름에 따라 정보 통신 기술의 주도권이 데이터로 이동하고 현존하는 데이터 양이 1ZB를 초과함에 따라 제타(Zeta) 시대에 돌입하였고, 이에, 정보 통신 기술의 주도권이 데이터(data)로 이동하게 되었다. 또한 모바일 시대로부터 스마트 시대로 시대의 흐름이 옮아감에 따라, 빅데이터는 미래의 경쟁력과 가치 창출의 원천이 되어가고 있다.

- 특징
  - 기존의 데이터 처리 응용 소프트웨어가 처리하기에 매우 거대(large)하거나 복잡한 (complex) 데이터 집합
  - 데이터를 취급하기 위한 전혀 다른 새로운 방법론의 필요성 대두
  - 정보 통신 기술의 주도권 ⇨ “데이터(data)”로 이동
  - 제타(Zeta) 시대에 돌입
  - 현존하는 데이터의 양이 1ZB를 초과
  - 모바일 시대 ⇨ 스마트 시대에 중요성 증대
  - 미래의 경쟁력과 가치 창출의 원천

● 빅데이터의 역사

빅데이터는 PC 시대, 인터넷 시대, 모바일 시대, 스마트 시대를 거침에 따라 그 중요성이 점차로 증대되고 있다. 과거에 비하여 데이터의 흐름이 증가하였고, 데이터를 다루는 플랫폼 또한 혁신적으로 변화하고 있다. 각 시대의 특징은 다음과 같다:

- PC 시대
  - 데이터베이스 개념 정립, PC통신 시작, 데이터의 규모가 작고 교류가 적음
- 인터넷 시대
  - 초고속 인터넷의 도입, 포털 서비스, 데이터의 규모 및 절대적 교류량 증가
- 모바일 시대
  - 모바일 인터넷 혁명, 소셜 네트워크, 빅데이터 개념 정립 및 문제점 대두

- 스마트 시대
  - 인공지능(AI), 기계학습(ML), 딥러닝, 사물인터넷(IoT) 등 지능을 가지는 시대

● 빅데이터의 3가지 특성 (3V)

빅데이터는 특징짓는 3가지 특성인 규모(volume), 다양성(variety), 속도(velocity)는 빅데이터의 정의 그 자체라고 할 수 있다. 규모의 관점에서 디지털 정보량이 기하급수적으로 증가하고 정보량 증가에 따른 데이터 처리 수요가 급증하였다. 다양성 관점에서는 로그, SNS, 소비 등 데이터의 종류가 증가하였고, 이에 따라 다양한 데이터에 대한 수용 및 처리 방법론이 필요해지고 있다. 마지막으로 속도 관점에서는 사물인터넷(IoT), 스트리밍 등 실시간 정보에 대한 요구와 속도가 증대되었고, 이에 따라 데이터를 빠르게 처리하고 분석하는 플랫폼이 필요해지고 있다.

- 규모Volume
  - 디지털 정보량이 기하급수적으로 급증⇧
  - 정보량 증가에 대한 데이터 처리 수요 증가
- 다양성Variety
  - 로그, SNS, 소비 등 데이터 종류 증가⇧
  - 다양한 데이터에 대한 수용 및 처리 방법론 필요
- 속도Velocity
  - IoT, 스트리밍 등 실시간 정보와 속도⇧
  - 데이터를 빠르게 처리하고 분석하는 플랫폼 필요

● 빅데이터의 새로운 특성

한편, 빅데이터를 특징짓는 새로운 4가지 특성인 정확성(veracity), 가치(value), 가변성(variability), 시각화(visualization)는 또다른 중요한 특성들이라 볼 수 있다. 정확성 관점에서는 거대 데이터에서는 신뢰성이 부족할 수 있어, 다양한 품질 문제에 대처하기 위한 방법론의 필요성이 대두되었다. 가치의 관점에서는, 트렌드, 감정, 진실성, 진정성, 개인 취향 등 변화하는 대규모 데이터 시대에 가치의 관점이 증대되고 있다. 가변성 관점에서는 맥락에 따라 의미가 변화함으로써 데이터의 본래 의미를 찾기 위한 방법론이 필요해지고 있다. 마지막으로 시각화는 데이터의 분석으로부터 얻어낸 결론에 대한 표현으로, 사용자의 이해도를 고려한 효과적인 방법론의 필요성이 증대되고 있다.

- 정확성Veracity
  - 거대 데이터는 신뢰성이 부족할 수 있음
  - 수집한 데이터의 다양한 품질에 대처하는 방법론
- 가치Value

- 트렌드, 감정, 진실성, 진정성, 개인의 취향 등
- 변화하는 대규모 데이터 시대에 가치의 필요성 증대

- 가변성Variability

- 맥락(context)에 따라 의미(meaning)가 변화
- 데이터의 본래 의미를 찾기 위한 방법론 필요

- 시각화Visualization

- 데이터의 분석으로부터 얻어낸 결론에 대한 표현
- 사용자의 이해도를 고려한 효과적인 방법론 필요

● 빅데이터의 종류

빅데이터는 정형 데이터, 반정형 데이터, 비정형 데이터로 구분할 수 있다. 정형 데이터는 고정형 필드에 저장된 데이터로, 관계형 데이터베이스나 스프레드 시트가 대표적인 예이다. 반정형 데이터는 메타 데이터, 스키마 기반 데이터 등이다. 마지막으로 비정형 데이터는 정형 데이터나 반정형 데이터의 범주에 들지 않는 데이터로, 텍스트 문서, 멀티미디어 콘텐츠 등이 그 대상이 될 수 있다.

- 정형 데이터Structured Data

- 고정형 필드(field)에 저장된 데이터
- 예) 관계형 데이터베이스(RDBMS), 스프레드시트 등

- 반정형 데이터Semi-Structured Data

- 메타데이터, 스키마를 이용하여표현되는 데이터
- 예) XML, HTML 등

- 비정형 데이터Unstructured Data

- 정형/반정형이 아닌 모든 데이터
- 예) 텍스트 문서, 멀티미디어 콘텐츠
- 가장 큰 증가⇧를 보이는 데이터 유형

2. 빅데이터 방법론

● 빅데이터의 처리 과정과 기술

빅데이터는 생성, 수집, 저장, 처리, 분석, 표현 과정을 거쳐 필요한 정보를 추출해낼 수 있다. 빅데이터 전문가라면 빅데이터의 처리 과정을 기억해두는 것이 매우 중요하다.

생성 → 수집 → 저장 → 처리 → 분석 → 표현

●

● 생성

데이터가 처음 생성되는 위치에 따라 내부 데이터와 외부 데이터로 구분할 수 있다. 내부 데이터는 로컬 환경에 저장되어 외부와 교류가 없는 데이터이며, 데이터베이스, 파일 관리 시스템 등을 예로 들 수 있다. 외부 데이터는 네트워크에서 교류를 통하여 발생하는 데이터로, 텍스트, 멀티미디어 콘텐츠, 스트림 등을 그 예로 들 수 있다.

- 내부 데이터
  - 로컬 환경에 저장되어 교류가 없는 데이터
  - 데이터베이스, 파일 관리 시스템 등
- 외부 데이터
  - 네트워크에서 교류를 통해 발생하는 데이터
  - 텍스트, 멀티미디어 콘텐츠, 스트림 등

● 수집

생성된 빅데이터는 수집(collection) 과정을 거쳐 수집하게 된다. 데이터의 수집 과정은 로그, 센싱, ETL 등을 통하여 수집할 수 있다. 로그는 시스템의 내부 활동 기록을 수집하는 것이며, 센싱은 각종 센서를 통하여 데이터를 수집하는 과정이다. 마지막으로 ETL은 추출(extraction), 변환(transformation), 적재(loading)의 과정으로, 데이터 웨어하우스의 구성이 그 예이다.

- 로그 (Log) : 시스템 내부 활동 로그 수집
  - 크롤링 (Crawling) : 인터넷 로봇을 사용한 데이터 수집
- 센싱 (Sensing) : 각종 센서를 이용한 수집
- ETL (Extraction, Transformation, Loading)
  - 소스 데이터의 추출, 변환, 적재
  - 데이터 웨어하우스(data warehouse)

● 저장

수집된 빅데이터는 용도에 맞는 적절한 저장소에 저장하는 것이 필수적이다. 데이터는 서버, 스토리지, NoSQL 등에 저장할 수 있다. 서버는 언제 어디서든 데이터에 효과적으로 접근하기 위한 장치이다. 스토리지는 데이터를 저장하는 전통적인 매체로, 자기 디스크, 광학 디스크 등이 있다. 마지막으로 NoSQL은 기존의 관계형 데이터베이스를 탈피하여 특히 비정형 데이터 관리에 특화되어 있는 데이터베이스의 한 형태이다.

- 서버 (Server)
  - 데이터에 효과적으로 접근하기 위한 장치
- 스토리지 (Storage)
  - 데이터를 저장하는 매체
- NoSQL 데이터베이스
  - 비정형 데이터 관리에 특화된 데이터베이스

● **처리**

저장된 빅데이터는 분석에 앞서 처리(processing) 과정을 거친다. 데이터 처리를 위해서는 다양한 플랫폼을 고려할 수 있으며, 맵리듀스, R, Hadoop, MATLAB 등이 대표적인 플랫폼이다.

- 맵리듀스 (MapReduce)
  - 분산 병렬 컴퓨팅에서 대용량 데이터를 처리하기 위한 소프트웨어 프레임워크
- R, Hadoop, MATLAB 등
  - 데이터 분석을 효과적으로 수행하기 위한 다양한 처리 도구

● **분석**

처리된 빅데이터는 분석(analysis) 과정을 거쳐 정보를 추출할 수 있다. 전통적인 고전 통계 분석을 통하여 다양한 경향성을 추출할 수 있으며, 최근 대두되는 기계학습, NLP 등의 방법을 적용하여 빅데이터로부터 다양한 인사이트(insight)를 얻을 수 있다.

- 통계 분석 (Statistical Analysis)
  - 고전 통계, 확률 모델링 등으로 분석 수행
- 기계학습 (Machine Learning)
  - 인공지능(AI)의 한 갈래이며 반자동 또는 자동으로 데이터의 패턴 발견
- NLP (Natural Language Processing)
  - 인간의 언어 현상을 분석하는 인공지능

● **표현**

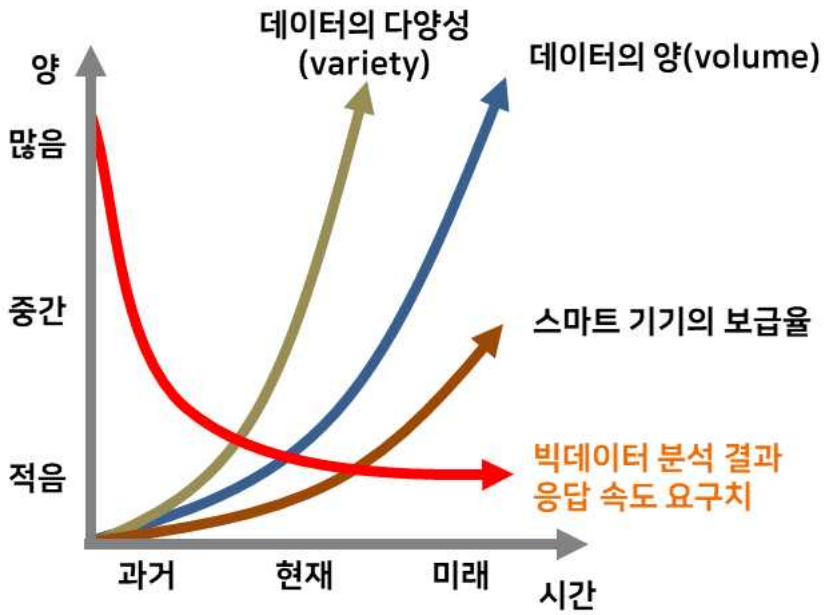
분석된 빅데이터의 정보는 적절한 형태로 표현되어야 그 가치를 발휘할 수 있다. 특히 시각화는 빅데이터로부터 알게 된 새로운 정보를 일반 대중들 또한 이해할 수 있도록 하여주는 매우 중요한 방법론이다.

- 시각화 (Visualization)
  - 다양한 도표와 그래픽으로 이해를 도움
  - 고전적인 표현 방식에 얽매이지 않고 형태, 색상, 매체, 구도 등을 변화하여 다양한 표현 적용

3. 빅데이터의 전망

● 빅데이터의 변화 추이

빅데이터는 다양한 관점에서 증가 혹은 감소를 하는 변화를 예상해 볼 수 있다. 먼저 '빅데이터'의 용어에서처럼, 과거에서 현재, 그리고 미래에 걸쳐 데이터의 양은 기하급수적으로 빠르게 증가할 것으로 예상한다. 데이터의 다양성은 과거에 비해 현재 매우 증가한 상황이고, 미래가 될수록 더 다양한 가치 추구 등으로 말미암아 대폭 증가할 것으로 예상된다. 스마트 기기는 현재 90% 이상이 보급되어 있는 상황이나, 해가 갈수록 보급률은 100%를 초과하게 될 것이며, 이에 따라 차별화된 빅데이터 기반의 정보 제공 및 서비스 전략이 필요할 것이다. 마지막으로 빅데이터의 분석 결과에 대한 반환 요구 시간은 점차로 감소하게 될 것이다.



● 빅데이터의 활용 분야

빅데이터는 공공 서비스, 과학 연구, 의료 서비스, 물류/유통, 제조 산업, 정보 통신 등 다양한 분야에서 활용될 것으로 전망된다.

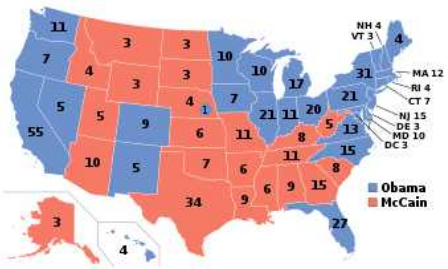
- 공공 서비스
  - 방대한 데이터를 국가적으로 활용 가능
  - 각종 자원 관리, 스마트 그리드, 재난 방재 등
- 과학 연구
  - 데이터로부터 새로운 의미 발견 가능
  - 데이터를 표현하는 새로운 방법론 적용
- 의료 서비스
  - 의료 데이터의 효율적, 효과적 수집 및 공유
  - 진단, 처방, 시술, 수술 등에 의료 혁명 예상
- 물류/유통
  - 데이터를 통해 소비자의 니즈(needs)를 파악
  - 보다 효과적인 물류 유통이 가능
- 제조 산업
  - 제품의 수율을 극대화하기 위하여 데이터를 활용
  - 불량률을 최소화할 수 있고 제조의 효율화 가능
- 정보 통신
  - 모바일 기기(스마트폰)의 보급 확대로 개인 데이터⇧
  - 개인화된 서비스 및 목표 마케팅 가능

● 빅데이터의 활용 사례

빅데이터는 현실의 정치, 경제, 문화, 과학 등 다양한 분야에 걸쳐 활용되어 오고, 또 미래에도 다양한 발상으로 활용될 것으로 전망된다.

- 정치
  - 유권자 DB에서 유권자를 분류하고 성향 파악
  - 소셜 미디어를 통하여 유권자 정보 수집
  - 유권자 별 맞춤형 선거 전략으로 효과적인 선거
  - 예) 2008년 미국 대통령 선거, 대한민국 제19대 총선

- 경제



2008년 미국 대통령 선거 결과

대한민국 제19대 총선

- 고객의 구매나 활동 기록을 분석하여 취향 분석
- 개인화 된 추천 시스템(recommender system)

- 문화

- 정형 및 비정형 데이터를 수집 및 분석
- 효율적인 팀의 운영을 가능케 함

- 과학

- 통계학의 비약적인 발전
- 유전자의 분석 및 발현과 조절에 획기적인 변화

● 빅데이터의 미래 전망

빅데이터는 미래에 데이터 혁명(data revolution)을 일으켜, 전 분야에 걸쳐 데이터가 발생하고 이로부터 정보를 얻어내 활용하는 문화가 확산될 것이다. 또한, 새로운 가치와 분야를 창출할 것으로 전망되고 있다.

- 데이터 혁명

- 정치, 경제, 사회, 문화, IT 등 데이터가 발생하는 전(全) 분야에 도입 ⇨ 사회 전반의 데이터 혁명

- 새로운 가치와 분야 창출

- 기존 데이터 뿐만 아니라 미래의 데이터로부터 새로운 의미를 찾아내고 가치와 분야 창출 기대

2강. 빅데이터 수집

본 강의에서는 빅데이터의 생성 이후 수집(collection)의 전 과정에 대하여 학습한다. 먼저 빅데이터의 생성과 다양한 관점의 구분을 학습하고, 빅데이터의 수집 절차에 대하여 학습한다.

1. 빅데이터의 수집 개요

빅데이터는 생성된 이래로 존재론적, 구성 등에 따라 다양하게 구분할 수 있다. 또한 빅데이터는 그 구성에 따라 다양한 유용성을 가지고 있다.

● 빅데이터의 생성

데이터는 관찰 및 측정을 통하여 획득할 수 있고, 가공되지 않은 상태이며, 단순한 사실이나 결과이다. 반면, 정보는 데이터를 가공하여 얻은 실질적인 결과이며, 의사결정에 기여하는 형태이다.

- 데이터Data
  - 관찰 및 측정을 통한 획득
  - 가공되지 않은 상태
  - 단순한 사실이나 결과
- 정보Information
  - 데이터를 가공하여 얻은 결과
  - 의사결정에 기여

● 데이터의 존재론적 특징에 따른 구분

데이터는 존재론적 관점에서 볼 때 정량적 데이터, 정성적 데이터로 구분할 수 있다. 정량적 데이터는 계량 가능한 형태의 데이터이며, 정형, 비정형의 형태를 가지고 있다. 반면, 정성적 데이터는 추상적 형태이며, 비정형 데이터의 형태를 가지고 있다.

- 정량적 데이터Quantitative Data
  - 언어, 문자 등 계량 가능 형태
  - 정형·비정형 데이터 형태
- 정성적 데이터Qualitative Data
  - 언어, 개념 등 추상적 형태

- 비정형 데이터 형태

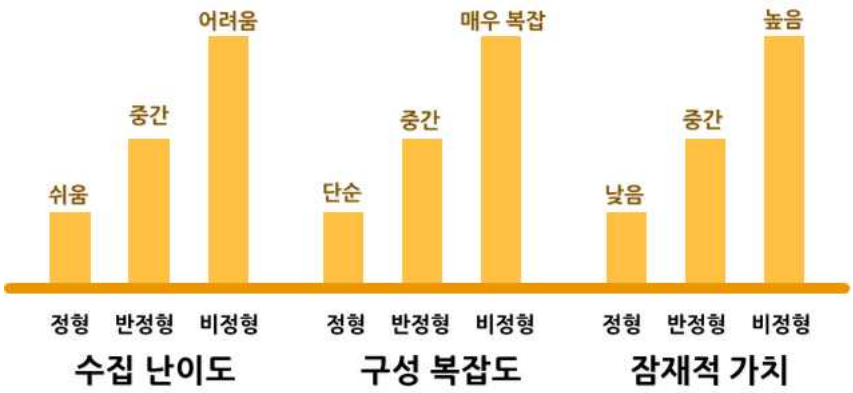
● 데이터의 구성에 따른 구분

데이터는 데이터의 구성에 따라 정형 데이터, 비정형 데이터, 반정형 데이터로 구분할 수 있다. 정형 데이터는 사전에 정의된 데이터의 모델이 존재한다. 한편, 비정형 데이터는 사전 정의된 데이터 모델이나 데이터 해석 방법론이 미약한 특징을 지니고 있다. 반정형 데이터는 정형 데이터와 비정형 데이터의 양쪽 모두의 특성을 일부 지니고 있는 데이터의 형태이다.

- 정형 데이터Structured Data
  - 사전 정의된 데이터 모델 존재
  - 최적화 된 자료구조 적용 가능
  - 예) 스프레드시트, DBMS
- 비정형 데이터Unstructured Data
  - 사전 정의된 데이터 모델이나 데이터의 해석 방법론이 미약
  - 예) 멀티미디어 콘텐츠, SNS
- 반정형 데이터Semi-structured Data
  - 정형-비정형 데이터의 중간
  - 예) HTML, XML, JSON, 로그

● 데이터의 구성에 따른 유용성

빅데이터는 수집 난이도, 구성 복잡도, 잠재적 가치에 따라 그 유용성이 달라진다. 정형 데이터보다 비정형 데이터의 수집 난이도가 높으며, 복잡도가 높고, 한편, 잠재적 가치 또한 비정형 데이터 쪽이 높다. 따라서 빅데이터에서는 앞으로 비정형 데이터의 취급 방법론이 대두될 것으로 예상된다.



● 빅데이터의 수집

빅데이터 수집은 시스템의 내외부에서 주기성을 가지고 필요한 형태로 데이터를 모으는 작업을 뜻한다. 빅데이터 수집을 통하여 유용한 데이터를 선택함으로써 산출물의 품질을 향상시킬 수 있으며, 최적의 방법론을 선택함으로써 수집 안정성을 극대화할 수 있고, 수집 소요 비용을 최소화할 수도 있다.

- 빅데이터 수집의 정의

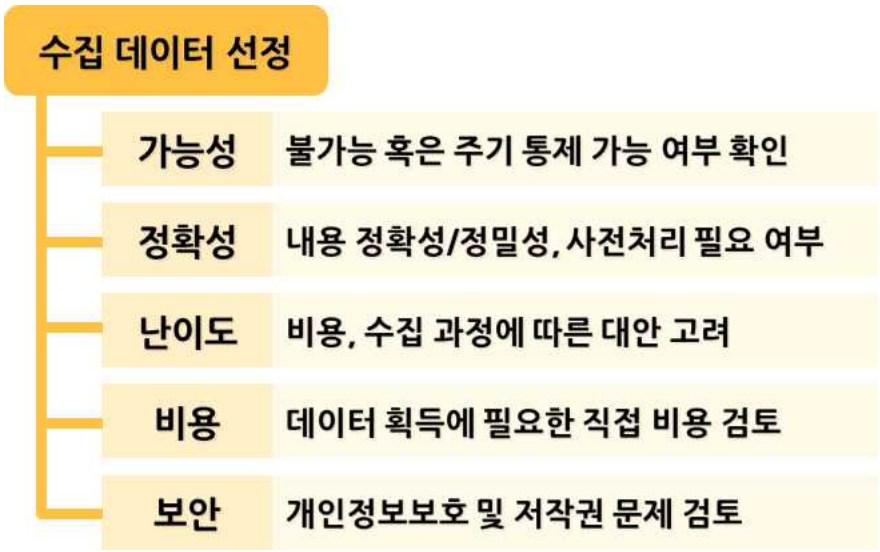
  시스템의 내외부에서 주기성을 가지고 필요한 형태로 데이터를 모으는 작업

- 빅데이터 수집의 역할
  - 유용한 데이터를 선택 ⇨ 산출물 품질 향상↑
  - 최적의 방법론 선택 ⇨ 수집 안정성 극대화↑
  - 수집 소요 비용 최소화↓

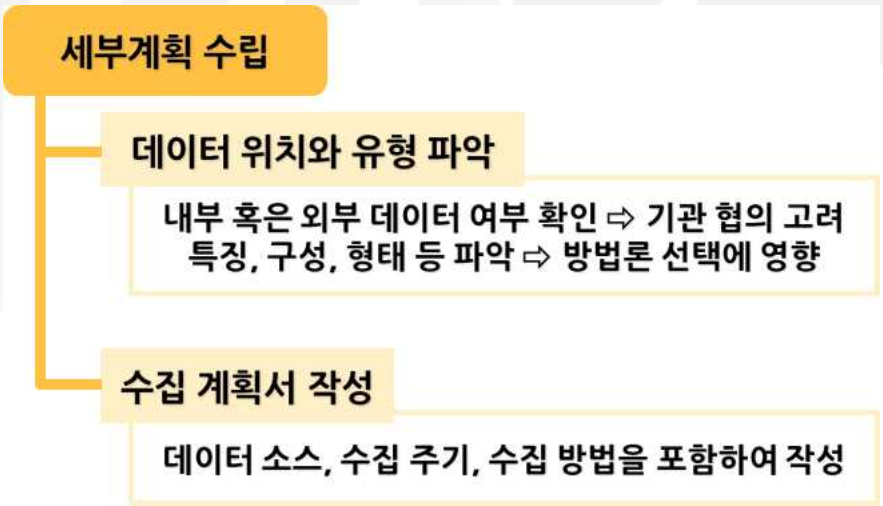
● 빅데이터 수집 절차 설계

빅데이터는 수집 데이터를 선정하고, 세부 계획을 수립하며, 테스트 수집을 진행하고, 본격적인 수집을 진행하는 절차를 가진다.

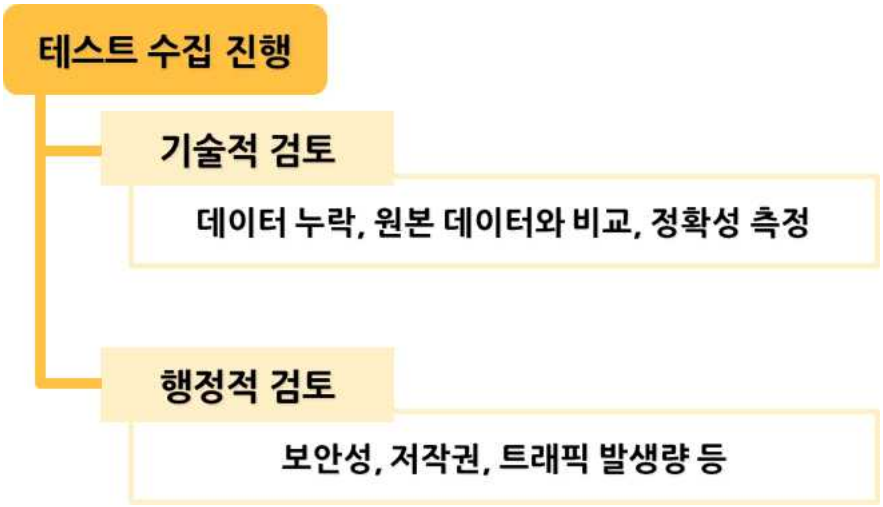
수집 데이터 선정 과정은 가능성, 정확성, 난이도, 비용, 보안 등의 관점에서 검증이 필요하다. 수집 데이터의 선정 가능성은 불가능 또는 주기의 통제 가능 여부를 확인하는 것이다. 수집 데이터의 선정 정확성은 내용의 정확성, 정밀성, 그리고 사전처리 필요 여부를 확인한다. 수집 데이터 선정의 난이도는 비용, 수집 과정에 따른 대안을 고려하는 것이다. 수집 데이터 선정의 비용은 데이터 획득에 필요한 직접적인 비용이 어느 정도인지 검토하는 것이다. 마지막으로 수집 데이터 선정의 보안성은 데이터 선정 대상의 개인정보 관련 문제 발생 여부를 검토하는 것이다.



수집 데이터를 선정한 이후에는 수집의 세부 계획을 수립하는 것이 필요하다. 데이터의 위치와 유형을 파악하고, 수집 계획서를 작성하는 것이 필요하다.



다음으로 본격적인 수집에 앞서 테스트 수집의 진행이 필요하다. 테스트 수집 과정 중 기술적인 검토와 행정적인 검토를 수행하게 된다. 기술적으로는 데이터 누락, 원본 데이터와의 전수 비교, 정확성 측정 등의 과정이 필요하다. 행정적으로는 보안성, 저작권 관련 문제, 트래픽 발생량 등의 고려가 필요하다.



2. 빅데이터의 수집 방법론

이번에는 본격적인 빅데이터의 수집 방법론에 대하여 학습한다. 빅데이터의 수집 계획서를 작성하는 데에 필요한 요소에 대하여 학습하고, 빅데이터의 다양한 수집 도구에 대하여 학습한다. 또한, 빅데이터의 수집을 자동화하기 위한 자동화 수집 기술에 대해서도 알아본다.

● 빅데이터 수집 계획서

빅데이터를 수집하기에 앞서, 빅데이터 수집 계획서를 작성하는 것이 필요하다. 빅데이터 수집 계획서에는 데이터 소스(원천), 수집 주기, 수집 방법 등이 정확히 기술되어 있어야 한다. 그 요소는 다음과 같다:

- 데이터 소스
  소스 위치, 형태, 인터페이스, 실무자, 협약 상세
- 수집 주기
  주기시간(규칙성) 또는 실시간(불규칙성), 데이터/트래픽량
- 수집 방법
  수집 기술, 사전/사후처리(pre/post processing), 대안

● 빅데이터 수집 도구

빅데이터의 수집에는 다양한 도구를 활용할 수 있다. 이러한 도구는 인적 자원, 자동화 도구로 구분할 수 있다. 인적 자원을 활용할 경우 인적 자원 비용이 발생하며, 오해석, 오차 등의 문제에 대한 대비책을 마련하여야 한다. 반면, 자동화 도구를 사용할 경우, 대부분의 과정에서 인간의 개입이 거의 없으며, 인적 자원 비용을 최소화하는 것이 가능하며, 데이터

원천 형태에 따라 적용이 불가능할 수 있으므로 주의하여야 한다.

- 인적 자원 활용Human Resource
  - 사람을 통하여 데이터 수집
  - 인적 자원 비용 발생
  - 오해석 또는 오차 등의 문제점
- 자동화 도구 사용Automatic Data Crawler
  - 대부분 과정에 사람 개입 없음
  - 인적 자원 비용 최소화 가능
  - 데이터 원천의 형태에 따라 적용이 불가능할 수도 있음

● **빅데이터 자동화 수집 기술**

빅데이터의 자동화 수집 기술은 주로 컴퓨팅 환경에서 이루어진다. 네트워크 수집, 로그/센서를 통한 수집 등이 대표적인 예이다. 네트워크를 통하여 수집할 경우, 크롤링 또는 OpenAPI를 사용할 수 있다. 로그/센서를 통하여 수집할 경우 로그를 기록하거나 센서로부터 유입되는 값을 기록하는 형식으로 데이터 수집이 가능하다.

- 네트워크 수집
  - 크롤링(Crawling) : 사전 정의 패턴에 따라 정해진 네트워크 지점의 데이터 수집
  - OpenAPI : 데이터 배포자 제공 인터페이스
- 로그/센서 수집
  - 로그(Log) 수집 : 작동 또는 이용 패턴의 기록
  - 센서(Sensor) 수집 : 센서 장치를 이용한 기록

3. 빅데이터의 수집 사례

빅데이터는 JSON, Flume, Chukwa, SQOOP, OpenRefine, Protocol Buffers 등 다양한 수집 플랫폼과 사례를 가지고 있다. 각 플랫폼의 주요 특징은 다음과 같다.

● JSON (JavaScript Object Notation)

- XML 유사 데이터 정형화 방식
- 인터넷 상의 데이터 송수신 방식
- 텍스트 형태, 작은 용량, 빠른 변환 속도

- 프로그래밍 언어 또는 플랫폼 독립적

- **Flume(플럼)**
  - 2010년 Cloudera 개발, 로그 데이터 수집기
  - 분산 데이터 통합 가능, 안정성 가용성 높음

- **Chukwa (척와)**
  - 2008년 Yahoo 개발, 로그 데이터 수집기
  - 아파치 하둡 기반, 실시간 분석 가능

- **SQOOP (스쿱)**
  - SQl-to-hadOOP, 다양한 DBMS 벤더 호환
  - DBMS, 하둡, NoSQL 간 데이터 연동에 적용

- **OpenRefine (오픈 리파인)**
  - 2010년 Google의 오픈 프로젝트
  - 데이터 정제 도구 : 오류 수정, 데이터 정리
  - 데이터 연계 API 및 워크플로우 기능 제공

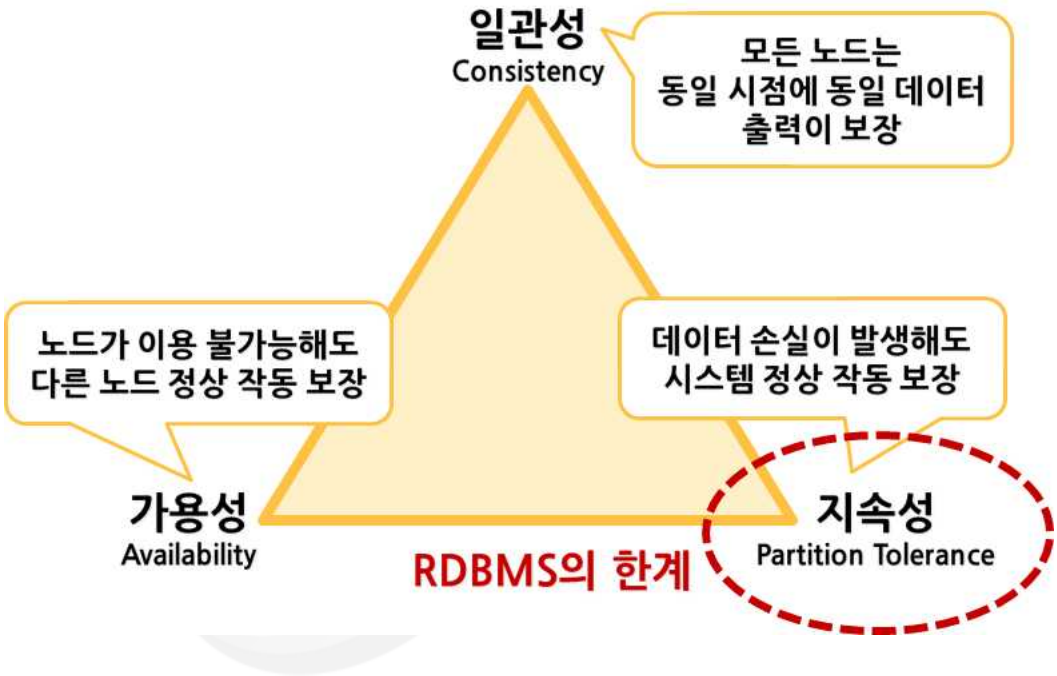
- **Protocol Buffers (프로토콜 버퍼)**
  - Google의 오픈소스 직렬화 라이브러리
  - 다양한 플랫폼 간 통신 가능

3강. 빅데이터 저장소

1. 빅데이터 저장소 개요

● CAP 이론

빅데이터의 저장소를 다루기에 앞서, 일반적인 저장소에서 다루는 CAP 이론의 세 가지 요소인 일관성(consistency), 가용성(availability), 지속성(partition tolerance)에 대한 이해가 필수이다. 기존의 RDBMS는 지속성의 관점에서 충족하지 못하였으나, 최근 빅데이터 시대가 대두되며 지속성을 만족하는 빅데이터 저장소가 필수가 되어가고 있다.



- 일관성Consistency
  - 모든 노드는 동일 시점에 동일 데이터 출력이 보장
- 가용성Availability
  - 노드가 이용 불가능해도 다른 노드 정상 작동 보장
- 지속성Partition Tolerance
  - 데이터 손실이 발생해도 시스템 정상 작동 보장
  - RDBMS의 한계

● RDBMS의 대안

앞서 설명한 지속성의 관점에서 한계가 있는 관계형 데이터베이스의 대안으로, 하둡,

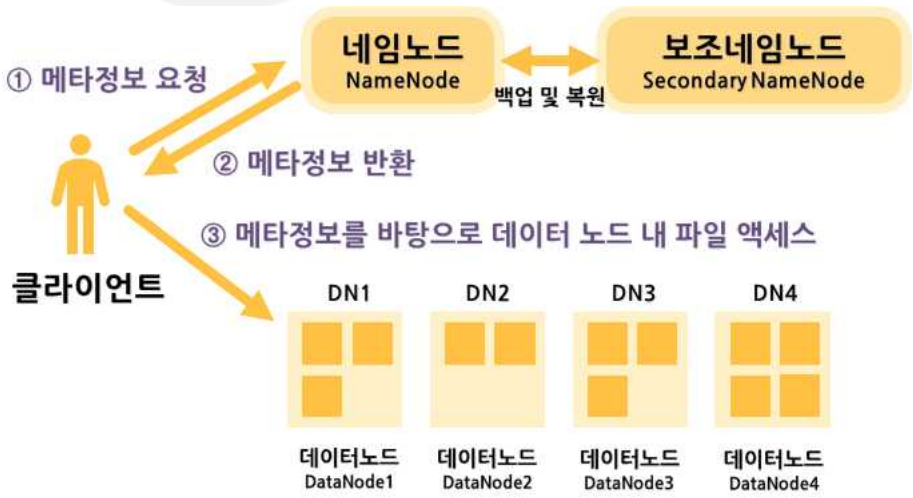
NoSQL, 레디스, 우지 등의 다양한 대안 저장소가 등장하고 있다. 그 특징은 다음과 같다:

- 하둡Hadoop
  - 하둡 분산파일시스템(HDFS)
  - 대용량 데이터 처리에 용이
- NoSQL
  - 비관계형 데이터 저장소
  - 데이터의 손실에 대처 가능
- 레디스Redis
  - NoSQL의 일종, 키-값 구조 저장소
  - 인메모리 저장소 구조
- 우지Oozie
  - 자바 서블릿 컨테이너 기반 작업 엔진
  - DAG 표현 기반 작업(job) 제어

2. 빅데이터 저장소 설계

● 하둡 분산 파일 시스템 (HDFS)

- 구성요소
  - 네임노드 서버 : 파일의 위치(iNode), 메타정보 관리, 클라이언트 요청
  - 보조 네임노드 서버 : 네임노드 서버의 백업 역할, 파일 시스템 복구
  - 데이터 노드 서버 : 고정된 크기의 블록 단위로 데이터를 나누어 저장



● NoSQL (Not-Only-SQL)

빅데이터 저장소 플랫폼으로 NoSQL은 주목받는 플랫폼이다.

키밸류, 빅테이블, 도큐먼트 등 다양한 데이터 모델을 제공하며, 비정형 데이터에 대응 가능하다.

- 데이터 모델
  - 키밸류KeyValue
  - 순차적 키밸류Ordered KeyValue
  - 빅테이블Bigtable
  - 도큐먼트Document
  - 그래프Graph

각 데이터모델의 특징을 살펴보면 다음과 같다.

- 키밸류KeyValue
  - 특정 값을 고유키와 대응하여 스키마 없이 데이터를 저장하는 유형
- 순차적 키밸류Ordered KeyValue
  - 키밸류 쌍을 순차적으로 저장, 연속성을 부여함으로써 영역 스캔 효율↑
- 빅테이블Bigtable
  - 테이블 형식(Tabular), 2~3단계까지 재귀적 구조 형성 가능
- 도큐먼트Document
  - 객체 데이터베이스(ODBMS)의 파생형. 데이터 구조 깊이에 제한 없음
- 그래프Graph
  - 가변적인 데이터 노드 간 연결 구조제약없는 관계 형성 가능

하둡 시스템은 마스터-슬레이브와 라운드 테이블의 특성을 가지고 있다.

- 시스템 구성
  - 마스터-슬레이브
  - 라운드 테이블

하둡은 마스터 노드와 슬레이브 노드가 존재하여, 메타 정보와 실제 데이터의 저장이 분리되어 있다.

- 마스터-슬레이브Master-Slave

- 마스터 노드가 슬레이브/데이터의 메타 정보 관리
- 서버 추가/삭제 작업에 용이
- 마스터에 부하 증가↑

한편, 라운드 테이블은 해시 테이블 기반의 구조를 가지고 있으며, 트래픽 분산에 용이하다는 장점이 있다. 반면, 멤버 노드에 변동이 발생할 경우 트래픽이 증가할 수 있다는 단점을 가진다.

- 라운드 테이블Round Table
  - 해시테이블(HashTable) 기반 구조
  - 마스터 노드 없음 → 트래픽 분산 용이
  - 멤버 노드 추가/삭제 시 데이터 이동 →트래픽 증가

3. 빅데이터 저장소 관리

본 장에서는 빅데이터 저장소의 대표적인 플랫폼 중 하나인 하둡에서 사용하는 기본 명령어에 대하여 학습하도록 한다.

● 하둡 명령어

- ls : 파일(폴더) 조회
  - 파일(폴더)를 조회하는 명령

```
> hadoop fs -ls /
Found 1 items
drwxr-xr-x  - hadoop supergroup     0 2019-02-22 06:08 /user
```

- put : 파일 올리기
  - 로컬 파일을 HDFS에 저장하는 명령
- get : 파일 가져오기
  - HDFS의 파일을 로컬로 가져오는 명령

```
> hadoop fs -put l_myData.txt h_myData.txt
```

```
> hadoop fs -get h_myData.txt l_myData.txt
```

- cp : 파일 복사하기
  - HDFS의 파일을 HDFS 상에서 복사
- rm : 파일 삭제하기
  - HDFS의 파일을 삭제하기

```
> hadoop fs -cp myData.txt myData2.txt
```

```
> hadoop fs -rm myData2.txt
```

- chmod : 권한 변경
  - HDFS 상의 파일의 권한을 변경
- chown : 소유권 변경
  - HDFS 상의 파일의 소유권을 변경

```
> hadoop fs -chmod 700 myData.txt
Found 1 items
-rw-------  1 hadoop supergroup   573 2019-02-22 06:15 /user/hadoop/myData.txt
```

```
> hadoop fs -chown bjhan myData.txt
```

● MongoDB (몽고DB)

빅데이터 저장소의 또 하나의 큰 흐름으로 NoSQL을 들 수 있다. 이러한 NoSQL의 철학을 그대로 구현한 저장소 플랫폼 중 하나가 바로 MongoDB이다. MongoDB는 문서 지향 데이터베이스 및 더블 링크드 리스트 구조를 가지고 있다는 특징이 있다.

- MongoDB 특징
  - 문서 지향 데이터베이스
  - 더블 링크드 리스트 구조

문서 지향 데이터베이스의 특징으로, 문서와 배열의 개념을 도입하고 있다. 또한 복잡한 계층 관계를 단순한 하나의 레코드로 표현 가능하며, NoSQL 데이터베이스의 철학을 따르고 있다.

- 문서 지향 데이터베이스
  - 문서(document)와 배열(array)의 개념 도입
  - 복잡한 계층 관계를 하나의 레코드로 표현 가능
  - NoSQL 데이터베이스

한편, MongoDB는 더블 링크드 리스트 구조를 가지고 있어, 데이터의 순방향 및 역방향 탐색이 가능하다는 특성을 가지고 있다.

- 더블 링크드 리스크 구조
  - 데이터의 순방향-역방향 탐색 가능

MongoDB에서는 데이터베이스, 컬렉션, 익스텐트, 도큐먼트로 모든 정보가 표현된다.

- MongoDB 주요 개념
  - 데이터베이스Database
  - 컬렉션Collection
  - 익스텐트Extent
  - 도큐먼트Document

각 정보 표현의 특징은 다음과 같다:

- 데이터베이스Database
  - 컬렉션의 논리적/물리적인 집합
- 컬렉션Collection
  - 구조적/개념적으로 유사한 도큐먼트의 집합
- 익스텐트Extent
  - 데이터 저장을 위한 논리 단위
- 도큐먼트Document
  - 정렬된 키(key)-값(value)의 집합

● MongoDB 명령어

다음은 MongoDB에서 사용하는 주요 명령어들이다.

- use
  - 데이터베이스를 생성(사용)하는 명령
- dropDatabase( )
  - 현재 사용하는 데이터베이스를 삭제

```
> use mydatabase
switched to db mydatabase
```

```
> db.dropDatabase()
```

- createCollection( )
  - 컬렉션을 생성(사용)하는 명령
- drop( )
  - 컬렉션을 삭제하는 명령

```
> db.createCollection("test")
{ "ok" : 1 }
```

```
> db.test.drop()
true
```

- insert( )

  - 도큐먼트를 추가하는 명령

```
> db.test.insert(
... {"title":"Big Data", "author":"bjhan"}
... {"title":"IoT", "author":"bjhan"});
BulkWriteResult({
    "writeErrors" : [ ],
    "writeConcernErrors" : [ ],
    "nInserted" : 2,
    "nUpserted" : 0,
    ...
```

- remove( )

  - 도큐먼트를 제거하는 명령

```
> db.test.remove({"title":"Big Data"})
{ "_id" : ObjectId("…"),
"title" : "Big Data", "author" : "bjhan" }
```

4강. 빅데이터 분석 도구 R (1)

1. R의 이해와 설치

● R이란?

R은 통계 분석, 그래픽 표현, 보고 작성을 위한 프로그래밍 언어 및 소프트웨어 환경이다. R의 통계 분석으로는 선형 및 비선형 모델링, 통계 검정, 시계열 분석, 분류, 군집화 등의 작업이 가능하다. R의 그래픽 표현 및 보고 작성 기능을 이용하여 막대형 그래프, 원형 그래프, 3차원 그래프 등 다양한 출력이 가능하다. R은 다음과 같은 특징을 가지고 있다:

- 통계 분석, 그래픽 표현, 보고 작성을 위한 프로그래밍 언어 및 소프트웨어 환경
- 통계 분석
  - 선형 및 비선형 모델링, 통계 검정, 시계열 분석, 분류, 군집화 등의 기능
- 그래픽 표현 및 보고 작성
  - 막대형 그래프, 원형 그래프, 3차원 그래프 등 출력
- GNU GPL Version 2 라이선스에 의하여 관리
- Windows, Linux, Mac 등 다양한 운영체제 환경 지원

● R의 특징

다른 빅데이터 처리 도구와 차별화된 R의 특징은 다음과 같다:

- 효과적인 데이터 핸들링 및 저장소 기능
- 선형대수 연산에 적합한 연산자 제공
- 일관성 있으며 통합된 데이터 분석 도구
- 데이터 분석의 그래픽 및 출력 기능
- 견고하면서도 간결하고 효과적인 프로그래밍 언어
  (조건문, 루프, 사용자 정의 재귀 함수, 입출력 기능 등)

● R의 설치

R은 윈도우, 리눅스 등 다양한 운영체제에서 설치 가능하며, 그 절차는 다음과 같다:

- Windows 환경에서의 설치
  - 공식 홈페이지의 바이너리 파일을 통하여 설치 가능
- Linux 환경에서의 설치
  - 패키지 등을 통하여 설치 가능
- 설치 명령문
  - Dpkg 기반인 경우

```
$ apt-get install r-base r-base-dev
```

  - RPM 기반인 경우

```
$ yum install R
```

● RStudio : R의 통합개발도구(IDE)

한편, R의 기능 중 그래픽 유저 인터페이스(GUI) 기능을 보완한 것이 바로 RStudio이다. 처음 R을 이용하여 데이터 분석을 하는 빅데이터 전문가라면 RStudio를 이용하여 통합 개발 도구 환경에서 개발하는 것이 권장된다.

- 좀더 편리한 분석 환경을 위하여 그래픽 사용자 인터페이스 (GUI) 제공 필요성 대두
- 소스코드 편집기, 디버깅, 시각화 도구를 포함
- www.rstudio.com 을 통하여 다운로드 가능
- 데스크톱 버전 : 오픈소스 에디션과 상업 라이선스 버전
  - 일반 사용자의 경우 오픈소스 에디션으로 충분
- 서버에서 구동되는 통합개발도구 및 실험 환경도 제공

RStudio의 특징은 다음과 같다.

- 특징 (오픈소스 에디션의 경우)
  - 소스코드 편집기를 이용하여 R의 명령문을 실행 가능
  - 소스코드 편집기에서는 문법 하이라이트, 자동완성,들여쓰기 등의 다채로운 기능 제공
  - 프로젝트와 작업 디렉터리의 관리 기능 제공
  - 통합된 도움말 및 문서 기능 제공

RStudio는 메뉴, 도구 바, 소스코드 편집 탭, 콘솔 탭, 환경 탭, 파일 탭, 플롯 탭 등으로 구성되며, 각 기능의 특징은 다음과 같다.

- 메뉴(Menu)
  - 파일, 편집, 코드, 보기 그래프(plots),세션, 빌드, 디버그, 프로파일, 도구,도움말 등의 기능에 접근 가능
- 도구 바(toolbar)
  - 자주 쓰는 기능을 아이콘으로 정의하여 편리하게 접근할 수 있도록 한 UI
  - 사용자가 직접 원하는 기능을 등록하여 사용 가능
- 소스코드 편집 탭 (Editor)
  - R 언어로 작성하는 소스코드를 입력
  - 문법(syntax)에 따른 하이라이트 기능
  - 자동 완성(auto completion) 기능
  - 자동 들여쓰기(auto-indent) 기능
- 콘솔 탭 (Console)
  - R 언어로 된 명령문을 직접 입력하여 실행 가능
  - 입력한 명령문의 실행 결과는 콘솔창으로 실시간 출력
  - 명령문을 이용하여 그래프 출력, 파일 입출력 가능
- 환경 탭 (Environment)
  - 현재 환경에서 정의된 변수의 일람을 볼 수 있는 곳
  - 변수의 이름, 변수의 값 등을 볼 수 있음
- 파일 탭 (Environment)
  - 시스템의 파일 읽기, 저장, 삭제 등
  - 디렉터리 생성, 변경, 삭제 등
- 플롯 탭 (Plots)
  - 그래프 명령어를 통한 결과가 나타나는 탭
  - 그래프의 설정을 직접 바꿀 수 있는 GUI 제공
  - 출력된 그래프를 추출(export) 가능

2. 기본 문법

R은 일종의 프로그래밍 환경이므로, 프로그래밍 언어로서의 기본적인 문법에 대하여 숙지하는 것이 필요하다. 여기서는 대입(할당) 연산자의 사용 방법과 주석의 사용 방법에 대하여 학습한다.

● 대입(할당) 연산자의 사용

- <- 연산자 또는 -> 연산자를 이용하여 변수에 값을 대입

```
# 변수 a에 3을 대입
a<-3
```

```
# 변수 b에 a+3의 결과를 대입
a+3->b
```

```
# 변수 l에 리스트를 생성하여 대입
l<-list("pizza", 3, 119.2)
```

- 주석의 사용

  - 주석(comments) : 실제로 실행되지 않는 코드의 메모

  - # 기호를 이용하여 주석 정의 가능

```
# 아래 명령문은 연산의 가능성을 알아보기 위한 부분임
a<-3
b<-4
c<-a+b
```

3. 변수의 사용

R에서는 다양한 데이터를 변수로서 다루게 된다. 여기서는 변수의 명명법, 변수 관련 함수의 용법에 대하여 숙지하도록 한다.

- 변수 (Variables)

  - 모든 변수는 문자, 숫자, 점(.), 밑줄 문자(_)만 사용 가능

  - 첫번째 문자는 숫자와 밑줄 문자(_)로 시작할 수 없음

  - 첫번째 문자가 점(.)인 경우 ⇨ 두번째 문자는 숫자 외 사용

- 변수 관련 함수

- class( ) : 변수의 데이터형을 알아내기 위한 함수

```
# 변수 a의 데이터형을 알아내기 위함
> class(a)
[1] "character"
>
```

```
# 변수 b의 데이터형을 알아내기 위함
> class(b)
[1] "numeric"
>
```

- ls( ) : 현재 사용하고 있는 변수의 목록 출력

```
# 사용하고 있는 변수의 목록을 보려고 함
> ls()
[1] "a" "b" "c" "l" "v"
>
```

```
# 사용하고 있는 변수 중 글자 v를 포함한 변수
> ls(pattern="v")
[1] "v"
>
```

- rm( ) : 변수를 지우는 함수

```
# a 변수를 지우고자 함
> rm("a")
> ls()
[1] "b" "c" "l" "v"
>
```

```
# 사용하고 있지 않는 a 변수를 지우려고 하는 경우
> rm("a")
Warning message:
In rm(a) : object 'a' not found
>
```

5강. 빅데이터 분석 도구 R (2)

1. 데이터형

R은 빅데이터를 분석하는 도구로, 데이터를 읽어들여 처리하기 위한 다양한 데이터형을 제공하고 있다. 빅데이터 전문가로서 이러한 데이터형에 대하여 숙지하고, 적재적소에 데이터형을 사용하는 것이 필요하다. 본 장에서는 R에서 사용하는 데이터형에 대하여 학습하여 본다.

● R의 데이터형

R에서는 논리형, 숫자형, 정수형, 복소수형, 문자형, 원형 등의 데이터형을 제공한다.

- 논리형(Logical)
- 숫자형(numeric)
- 정수형(integer)
- 복소수형(complex)
- 문자형(character)
- 원형(raw)

각 데이터형의 특징과 용법을 살펴보면 다음과 같다.

● 논리형 (Logical)

- TRUE(참), FALSE(거짓)의 값을 가짐

```
# 논리 데이터형 테스트
> l <- TRUE
> class(l)
[1] "logical"
>
```

● 숫자형 (Numeric)

- 실수를 표현하기 위한 데이터형

```
# 숫자 데이터형 테스트
> n<-3.1415
> class(n)
[1] "numeric"
>
```

- **정수형 (Integer)**

  - 소수점이 없는 정수를 표현

```
# 정수 데이터형 테스트
> i<-1147L
> class(i)
[1] "integer"
>
```

- **복소수형 (Complex)**

  - 복소수를 표현하고자 할 때 사용

```
# 복소수 데이터형 테스트
> c<-3+5i
> class(c)
[1] "complex"
>
```

- **문자형 (Character)**

  - 문자 또는 문자열을 표현할 때 사용

```
# 문자 데이터형 테스트
> s<-"hello"
> class(s)
[1] "character"
>
```

- **원형 (Raw)**

  - 컴퓨터 시스템에서 표현하는 기본 형태

```
# 원형 데이터형 테스트
> r<-charToRaw("hello")
> r
[1] 68 65 6c 6c 6f
>
```

2. R객체 (R-Object)

R객체는 R에서 지원하는 특수한 형태의 객체(object)로, 자주 쓰이는 객체는 적재적소에 활용하는 것이 필수이다. 다음은 R에서 자주 쓰이는 R객체들이다.

- 자주 쓰이는 R객체들
  - 벡터 (Vectors)
  - 리스트 (lists)
  - 행렬 (matrices)
  - 배열 (arrays)
  - 요인 (factors)
  - 데이터 프레임 (data frames)

이후 R에서 쓰이는 R객체들의 정의와 간단한 예시들에 대하여 숙지하고 모두 실습해 보도록 하자.

● 벡터 (Vectors)

- 다수의 값을 담고 있는 R객체

- 내부 데이터는 한 가지 데이터형으로 통일

```
# 벡터 R객체 테스트
> v<-c("hello",3.14) #문자형과 숫자형 혼용
> class(v)
[1] "character"      #문자형으로 변화
> v
[1] "hello" "3.14"
>
```

● 리스트 (Lists)

- 서로 다른 유형의 데이터를 담을 수 있는 R객체

```
# 리스트 R객체 테스트
> l<-list("hello",3.14) #문자형과 숫자형 혼용
> class(l)
[1] “list"                #리스트형으로 취급
> v
[[1]]
[1] "hello“
[[2]]
[1] 3.14
>
```

- **행렬 (Matrices)**

  - 행과 열로 이루어진 2차원 데이터 집합

```
# 행렬 R객체 테스트
> m<-matrix(c(2,3,1,5),nrow=2,ncol=2)
> class(m)
[1] "matrix"
> m
     [,1] [,2]
[1,]    2    1
[2,]    3    5
>
```

- **배열 (Arrays)**

  - 다차원으로 구성된 데이터 집합

  - 차원의 설정에 따른 배열의 변화

    - 1차원으로 설정 ⇨ 배열 (Arrays)로 취급
    - 2차원으로 설정 ⇨ 행렬 (Matrix)로 취급
    - 3차원 이상으로 설정 ⇨ 배열 (Arrays)로 취급

```
# 배열 R객체 테스트
> ar1<-array(c(2,3,1,5),dim=c(4))
> ar2<-array(c(2,3,1,5),dim=c(2,2))
> ar3<-array(c(2,3,1,5),dim=c(1,2,2))
> class(ar1)
[1] "array"
> class(ar2)
[1] "matrix"
> class(ar3)
[1] "array"
> ar1
[1] 2 3 1 5
> ar2
     [,1] [,2]
[1,]    2    1
[2,]    3    5
```

```
> ar3
, , 1

     [,1] [,2]
[1,]    2    3

, , 2

     [,1] [,2]
[1,]    1    5

>
```

- 요인 (Factors)
  - "범주"라고도 부름
  - 데이터의 값(value)과 레벨(label)을 함께 표현
  - 객체가 구축되며 데이터가 자동으로 분석되므로 통계적 모델링과 분석에 유용

```
> v<-c("kim","kim","han","lee","lee","kim")
> vf<-factor(v)
> vf
[1] kim  kim  han  lee  lee  kim
Levels: han kim lee
> class(vf)
[1] "factor"
>
```

- 데이터 프레임 (Data Frames)
  - 표의 형태로 정리된 데이터 객체의 일종
  - 열(column)과 행(row)의 이름(name)을 지정 가능
  - 열마다 서로 다른 데이터형을 가질 수 있음
  - 숫자형(numeric), 요인(factor), 문자형(character)외의 다른 데이터를 저장할 수 없음
  - 열마다 같은 수의 데이터를 포함하여야 함

```
> v<-c("kim","kim","han","lee","lee","kim")
> vf<-factor(v)
> vf
[1] kim  kim  han  lee lee  kim
Levels: han kim lee
> class(vf)
[1] "factor"
> summary(df)
    gender      height          weight        age
 female:2   Min.   :159.0   Min.   :49   Min.   :25.00
 male  :1   1st Qu.:161.0   1st Qu.:51   1st Qu.:30.50
            Median :163.0   Median :53   Median :36.00
            Mean   :166.7   Mean   :65   Mean   :40.33
            3rd Qu.:170.5   3rd Qu.:73   3rd Qu.:48.00
            Max.   :178.0   Max.   :93   Max.   :60.00
>
```

2. 연산자

R에서는 다양한 연산자가 제공되어, 빅데이터 분석에서 사용하는 다양한 방법론을 구현하여 사용하는 것이 가능하다. R에서 제공하는 연산자들은 산술 연산자, 관계 연산자, 논리 연산자, 대입(할당) 연산자, 기타 연산자로 구분할 수 있다.

● 산술 연산자 (Arithmetic Operators)

산술 연산자는 덧셈, 뺄셈, 곱셈, 나눗셈 등의 사칙연산과 지수 연산 등을 수행하는 가장 기본이 되는 연산자이다.

연산자	사용법	연산자의 반환 결과
+	a+b	• 덧셈을 수행한 후 결과 반환
-	a-b	• 뺄셈을 수행한 후 결과 반환
*	a*b	• 곱셈을 수행한 후 결과 반환
/	a/b	• 나눗셈을 수행한 후 결과 반환
%%	a%%b	• 나눗셈을 수행한 후 나머지 반환
%/%	a%/%b	• 나눗셈을 수행한 후 몫 반환
^	a^b	• a의 b승을 계산

```
> 7+3   # 덧셈 연산 수행
[1] 10
> 7-3   # 뺄셈 연산 수행
[1] 4
> 7*3   # 곱셈 연산 수행
[1] 21
> 7/3   # 나눗셈 연산 수행
[1] 2.333333
> 7%%3  # 나머지 연산 수행
[1] 1
> 7%/%3 # 몫 연산 수행
[1] 2
> 7^3   # 지수 연산 수행
[1] 343
>
```

```
> a<-c(6,7) # 벡터 a 생성
> b<-c(4,3) # 벡터 b 생성
> a+b   # 성분별 덧셈
[1] 10 10
> a-b   # 성분별 뺄셈
[1] 2 4
> a*b   # 성분별 곱셈
[1] 24 21
> a/b   # 성분별 나눗셈
[1] 1.500000 2.333333
> a%%b  # 성분별 나머지 연산
[1] 2 1
> a%/%b # 성분별 몫 연산
[1] 1 2
> a^b   # 성분별 지수 연산
[1] 1296  343
>
```

- 관계 연산자 (Relational Operators)

관계 연산자는 의사결정문에서 주로 쓰이는 연산자로, 참과 거짓을 판별하기 위한 대소관계 비교 연산자, 값의 동일 여부를 검증하는 연산자 등이 제공되고 있다.

연산자	사용법	연산자의 반환 결과가 TRUE
>	a>b	• a가 b보다 큰 경우
>=	a>=b	• a가 b보다 크거나 같은 경우
<	a<b	• a가 b보다 작은 경우
<=	a<=b	• a가 b보다 작거나 같은 경우
==	a==b	• a와 b의 값이 서로 같은 경우
!=	a!=b	• a와 b의 값이 서로 다른 경우

```
> a<-c(2,3,4)            # 벡터 a 생성
> b<-c(3,3,3)            # 벡터 b 생성
> a>b  # 크다(>) 연산
[1] FALSE FALSE  TRUE
> a>=b # 크거나 같다(>=) 연산
[1] FALSE  TRUE  TRUE
> a<b  # 작다(<) 연산
[1]  TRUE FALSE FALSE
> a<=b # 작거나 같다(<=) 연산
[1]  TRUE  TRUE FALSE
> a==b # 같다(==) 연산
[1] FALSE  TRUE FALSE
> a!=b # 다르다(!=) 연산
[1]  TRUE FALSE  TRUE
>
```

- 논리 연산자 (Logical Operators)

논리 연산자는 R의 객체 내부의 각 성분별 비교 연산을 수행하고, 그에 따른 결과를 반환하는 연산자로, 이산수학에서 활용되는 논리곱, 논리합, 반전 등의 연산자가 제공된다.

연산자	사용법	연산자의 반환 결과
&	a&b	• a와 b의 각 성분별로 AND 연산 결과
|	a|b	• a와 b의 각 성분별로 OR 연산 결과
!	!a	• a의 논리 결과를 반대로 반환 (NOT)
&&	a&&b	• a와 b의 첫번째 원소에 대해 AND 연산 결과
||	a||b	• a와 b의 첫번째 원소에 대해 OR 연산 결과

```
> a<-c(TRUE, FALSE, FALSE)   # 벡터 a 생성
> b<-c(TRUE, TRUE, FALSE)    # 벡터 b 생성
> a&b    # 벡터 성분별 AND 연산
[1]  TRUE FALSE FALSE
> a|b    # 벡터 성분별 OR 연산
[1]  TRUE  TRUE FALSE
> !a     # 벡터 성분별 NOT 연산
[1] FALSE  TRUE  TRUE
> a&&b   # 벡터 첫번째 성분 AND 연산
[1] TRUE
> a||b   # 벡터 첫번째 성분 OR 연산
[1] TRUE
>
```

- 대입(할당) 연산자 (Assignment Operators)

대입 연산자는 변수와 변수, 혹은 상수와 변수 사이에 값을 대입 또는 할당하기 위한 연산자로, R에서 데이터를 불러오고 저장하며 중간 연산결과를 저장할 때 다양하게 활용하는 연산자들의 묶음이다.

연산자	사용법	연산자의 반환 결과
<- = <<-	a<-b a=b a<<-b	• b의 값을 a에 대입(할당)
->> ->>	a->b a->>b	• a의 값을 b에 대입(할당)

```
> a<-3
> a=3
> a<<-3
> a->3
Error in 3 <- a : invalid (do_set) left-hand
side to assignment
> 3->a
> 3->>a
>
```

- 기타 연산자 (Miscellaneous Operators)

그 외에 기타 연산자는 R에서 벡터를 생성하거나 성분의 존재 여부를 검증할 때 활용하는 다양한 연산자들이다.

연산자	사용법	연산자의 반환 결과
:	a:b	• 시작값은 a이고 끝값은 b보다 작거나 같은 벡터를 생성
%in%	a%in%b	• a가 b의 성분인 경우 TRUE를 반환하고, 그렇지 않은 경우 FALSE를 반환
%*%	a%*%b	• a와 b가 행렬 R객체인 경우, 행렬 간 곱셈을 수행

```
> 3:8     # 콜론(:) 연산자로 수열 벡터 생성
[1] 3 4 5 6 7 8
> "pizza"%in%c("pizza","hamburger") # 문자열 포함 여부 검정
[1] TRUE
> "cheese"%in%c("pizza","hamburger") # 문자열 포함 여부 검정
[1] FALSE
> a<-matrix(c(1,2,3,4),nrow=2,ncol=2) # 행렬 a 생성
> b<-matrix(c(4,3,2,1),nrow=2,ncol=2) # 행렬 b 생성
> a*b   # 성분별 곱셈
     [,1] [,2]
[1,]    4    6
[2,]    6    4
> a%*%b # 선형대수(행렬) 곱셈
     [,1] [,2]
[1,]   13    5
[2,]   20    8
>
```

6강. 빅데이터 분석 도구 R (3)

1. 의사결정 구조

R은 하나의 프로그래밍 환경으로, 의사결정 구조를 통하여 데이터의 형태에 따라 다양한 분기를 통한 처리를 하는 것이 중요하다. R에서는 if문, if~else문, switch문 등 다양한 문법을 통하여 의사결정 구조를 구현할 수 있도록 돕고 있다.

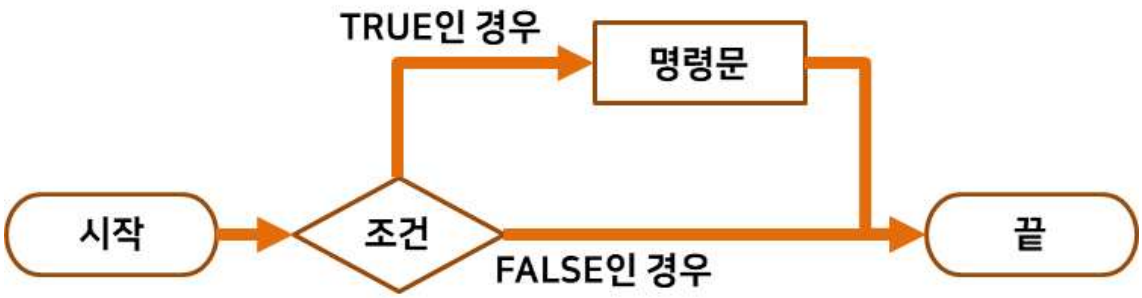
- 하나 이상의 조건을 평가(또는 테스트)하여 그 결과에 따라 다양한 흐름으로 명령문을 수행하는 구조
- R에서는 if문, if~else문, switch문을 제공

● 의사결정 구조의 용도

- if문 : 조건이 참일 때만 명령문을 수행
- if~else문 : 조건에 따라 별개의 명령문을 수행
- switch문 : 다양한 값의 조건에 따라 별개의 명령문을 수행

● if 문

if문은 조건이 참인 경우에만 명령문을 수행하는 의사결정 구조문이다. 가장 단순한 의사결정 구조문이나, R로 다루는 모든 프로그램 로직의 골격을 이루고 있다.



- 특징
  - 조건이 참(TRUE)인 경우에만 명령문 수행
  - 가장 단순한 의사결정 구조
- 문법

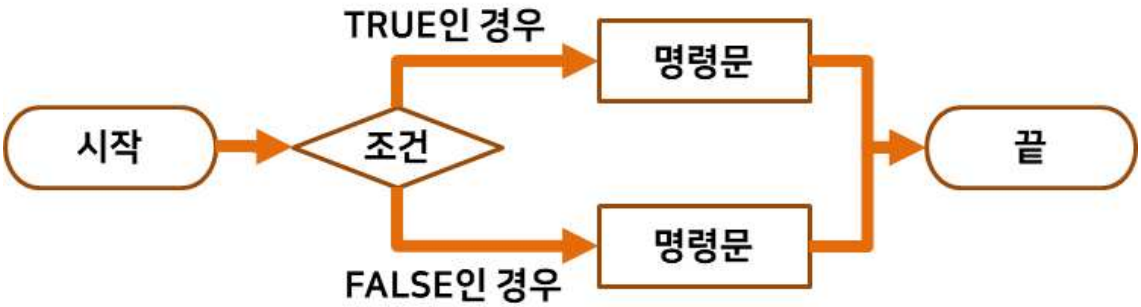
```
if (조건) {
   … # 조건이 참(TRUE)인 경우 수행하는 명령문
}
```

- 예시

```
> x<-36L
> if(is.integer(x)) {
+     print("x는 정수")
+ }
[1] "x는 정수"
>
```

- if~else문

if~else문은 참일 뿐만 아니라 거짓인 경우에도 처리하는 구문을 추가한 의사결정 구조문 형태로, 분기를 나누어서 작업을 처리할 때 주로 사용된다.



- 특징
  - 조건이 참(TRUE)인 경우와 거짓(FALSE)인 경우서로 다른 명령문을 수행하는 의사결정 구조
- 문법

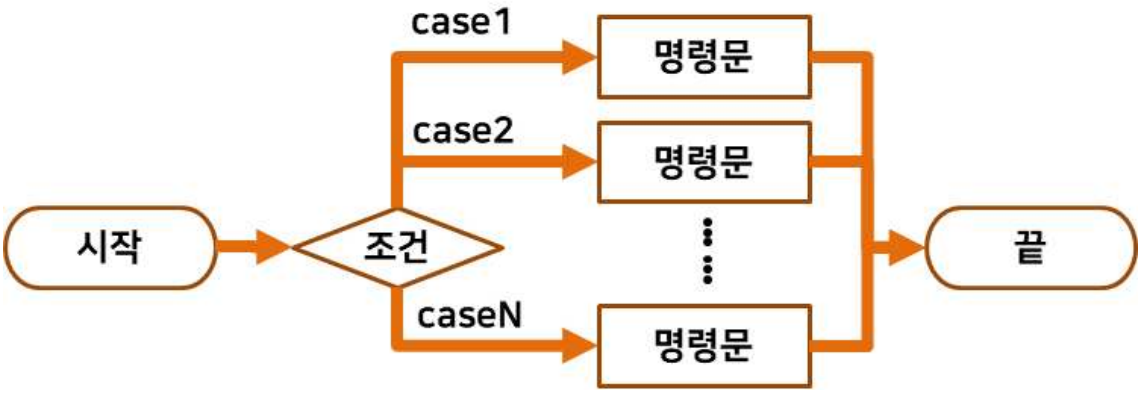
```
if (조건) {
   … # 조건이 참(TRUE)인 경우 수행하는 명령문
} else {
   … # 조건이 거짓(FALSE)인 경우 수행하는 명령문
}
```

- 예시

```
> x<-3.14
> if(is.integer(x)) {
+     print("x는 정수")
+ } else {
+     print("x는 정수가 아님")
+ }
[1] "x는 정수가 아님"
>
```

- switch문

switch문은 조건에 따라 다양한 분기로 명령문을 실행하는 구조를 가지고 있다. 참과 거짓이 아닌 값에 따른 다양한 분기를 가지는 의사결정 구조를 구현할 때 유용하다.



- 특징

  - 조건이 일치하는 경우(case)의 명령문을 실행
  - 조건이 다수로의 분기를 가질 때 유리
  - 다른 프로그래밍 언어와 달리 기본값(default)이 없음

- 문법

```
switch (조건, case1, case2, …, caseN)
```

- 예시

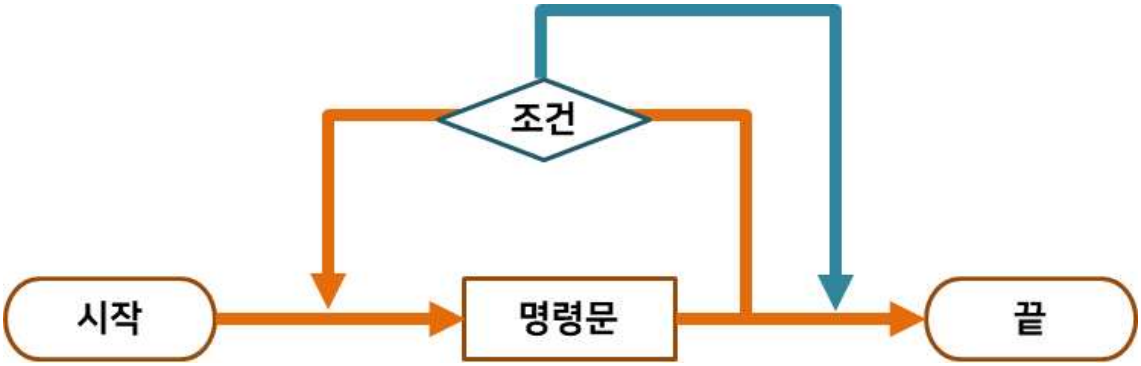
```
> x<-"han"
> switch(x, "kim"="김", "lee"="이", han="한")
[1] "한"
>
```

2. 루프

인간이 컴퓨터를 사용하게 된 것은 단순반복적인 작업을 컴퓨터를 통하여 진행하고자 하는 것이 그 큰 목적 중 하나이다. 프로그래밍 환경인 R에는 그러한 철학을 이어받아 반복문(루프) 또한 존재하고 있다. 본 장에서는 R에서 사용할 수 있는 다양한 루프문에 대하여 학습하여 본다.

- **정의**
  - 특정한 명령문을 조건에 따라 여러 번 실행하고자 할 때 사용하는 구조
- **종류**
  - repeat 루프
  - while 루프
  - for 루프
- **repeat 루프**

repeat문은 무한히 반복하여 명령문을 실행하는 구조이다. if 및 break문을 이용하여 반복구조에서 탈출할 수도 있다.



- 특징

  - 무한히 반복하여 명령문을 실행하는 루프 구조
  - if문과 break를 사용하여 반복에서 탈출할 수 있음

- 문법

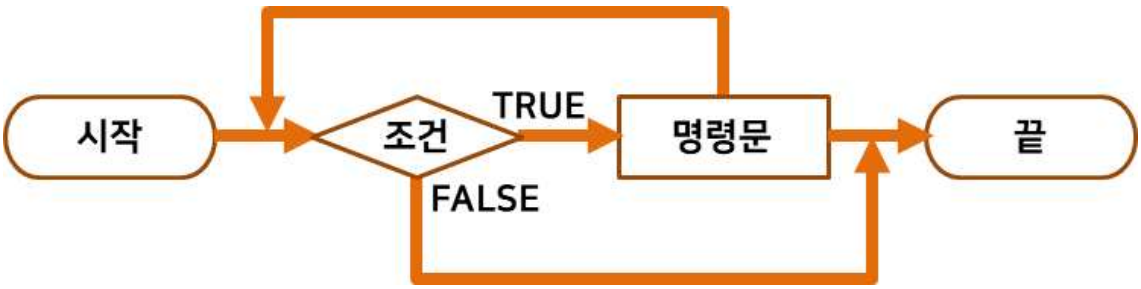
```
repeat {
  … # 반복하여 실행할 명령문
}
```

- 예시

```
> i<-1
> repeat {
+     i<-i+1
+     if(i==3) {    # break문이 없는 경우
+         print(i)  # 무한히 실행되므로
+         break     # if문으로 탈출 조건 적용
+     }
+ }
[1] 3
>
```

● while 루프

while문은 조건이 참인 경우에만 한정하여 동일한 명령문을 반복하여 실행하는 구조이다. 중간에 조건문의 조건이 바뀌게 되면 반복문을 탈출할 수도 있다.



- 특징
  - 조건이 참이면 동일한 명령문을 다시 실행하는 루프
  - 명령문의 실행 결과에 따라 조건이 거짓이 되면 반복문으로부터 탈출하는 것이 가능

- 문법

```
while (조건) {
  … # 조건이 참(TRUE)이면 반복하여 실행할 명령문
}
```

- 예시

```
> i<-1
> while(i<3) { # i가 3 미만인 경우 무수히 반복
+     i<-i+1
+ }
> i
[1] 3
>
```

● for 루프

- 특징
  - 벡터의 각 성분에 따라 반복하는 루프 구조
  - 벡터 성분의 위치(index)에 일일히 접근하지 않으므로 벡터 성분별 명령문을 수행하는 구조일 때 유리
  - 벡터 성분의 값을 대입하기 위한 변수 지정 필요

- 문법

```
for (value in vector) {
    … # 각 성분별 실행할 명령문
}
```

- 예시

```
> v<-c(2,3,5,7,10)
> s<-0
> for(ve in v) {
+     s<-s+ve
+ }
> print(s)
[1] 27
>
```

- **break문**
  - 특징
    - 현재 실행중인 루프문을 중단하고, 루프문 이후 명령문부터 실행하는 루프 제어 명령문
- **next문**
  - 특징
    - 현재 실행중인 루프문을 중단하고, 루프문의 처음으로 되돌아가 명령문을 실행하도록 흐름을 제어하는 루프 제어 명령문
  - 적용 방법
    - if문 등을 이용하여 사용하는 것이 일반적
  - next문의 예시

```
> v<-1:10
> for(ve in v) {
+     if( ve%%2 != 0 ) { # %% : 나머지 연산자
+         next
+     }
+     print(ve)
+ }
[1] 2
[1] 4
[1] 6
[1] 8
[1] 10
>
```

3. 함수

- **함수(function)**

함수는 특정한 작업을 수행하는 명령문과 구조를 나열한 집합으로, 특히 사용자 정의 함수를 사용하여 사용자가 직접 자신만의 기능을 구현하여 활용할 수 있다.

- 정의
  - 특정한 작업을 수행하는 명령문과 구조를 나열한 집합
- 기본 함수 (built-in functions)
  - R에서 기본으로 정의되어 있는 함수
- 사용자 정의 함수 (user defined functions)
  - 사용자가 자신의 목적에 맞게끔 함수명, 명령문,실행구조, 입력인자, 출력인자를 지정한 함수

● 함수 호출 (Function Calling)

R에서의 함수 호출 문법과 예시는 다음과 같다. 함수를 호출할 때에는 입력인자가 있는 경우 입력인자를 순서대로 나열함으로써 함수에 입력인자의 값을 전달할 수 있다. 한편, R에서만의 특이한 문법으로, 함수 입력인자명을 지정함으로써 특정 입력인자를 할당하는 것이 가능하며, 이는 실제의 입력인자 순서와는 무관하게 지정 가능하다.

- 문법

```
<함수명>(<입력인자1>,<입력인자2>,…)
```

```
<함수명>(입력인자명1=<입력인자1>,입력인자명2=<입력인자2>,…)
```

- 예시

```
mean(c(88,96,92)) # mean과 c함수의 호출
```

```
png(file="result.png") # 입력인자명을 지정하여 호출
```

● 사용자 정의 함수 생성 문법

R에서는 사용자 정의 함수를 생성할 수 있다. 사용자 정의 함수를 생성하ㅁ 위해서는 function 키워드를 사용하여 함수를 생성할 수 있다. 이때, function 키워드로 생성한 함수를 함수명에 대입하는 형태로 정의할 수 있다. 한편, 입력인자명을 지정할 수도 있으며, 입력인자명을 지정할 때에는 “<입력인자명>=<입력인자>”와 같은 구조를 따른다.

사용자 정의 함수를 생성할 때에는 반드시 함수명, function 키워드, 중괄호 등을 포함하여야 하며, 선택요소는 입력인자, 출력인자, 입력인자명 등이다. 즉, 극단적으로는 입력인자, 출력인자, 입력인자명이 모두 없는 사용자 정의 함수도 존재할 수 있다.

```
<함수명><-function((입력인자명1=)<입력인자1>, …) {
    … # 함수가 호출되면 실행할 명령문 및 구조
}
```

- 구성요소
  - 필수요소 : 함수명, function 키워드, 중괄호 등
  - 선택요소 : 입력인자, 출력인자, 입력인자명 등

- 예시

```
> myFunc<-function(v) { # v벡터를 받아 합산결과를 반환하는
+     s<-0              # 사용자 정의 함수 myFunc()
+     for(ve in v) {
+         s<-s+v
+     }
+     return(s)
+ }
> myFunc(c(2,3,7))  # 사용자 정의 함수의 호출
[1]  6  9 21
>
```

7강. 데이터 통계 분석 (1)

1. 확률과 통계이론

R은 데이터 처리 및 분석 도구로서, 처리와 분석을 위한 다양한 이론을 습득하고 적재적소에 적용하는 것이 중요하다. 본 강의에서는 그 출발점으로, 확률과 통계 이론에 대하여 다루고자 한다. 먼저 통계에 대하여 정의내리고, 모집단과 표본의 차이점에 대하여 이해하도록 한다. 다음으로, 확률에 대하여 정의내리고, 다양한 확률 분포에 대하여 학습하고자 한다.

● 통계 (Statistics)

통계는 표본을 통하여 모집단을 추정하기 위한 학문이다. 빅데이터는 거대한 표본이지만 그보다 더 거대한 모집단의 일부 데이터에 지나지 않으므로, 빅데이터에서도 통계의 이론은 그대로 적용된다. 즉, 표본을 수집하고 분석하는 고전적인 프로세스는 빅데이터에서 그대로 적용되므로, 다른 어떤 분야만큼이나 통계에 대한 이해는 필수라고 볼 수 있다.

- 정의
  - 표본을 통하여 모집단을 추정하기 위한 학문
- 빅데이터와 관계
  - 표본(=데이터)을 수집하고 분석하는 고전 과정
  - 빅데이터의 기반 학문

● 모집단과 표본

빅데이터는 하나의 거대한 표본이며, 빅데이터의 궁극적인 목적 중 하나는, 표본을 이용하여 모집단을 이해하려고 하는 것이다. 고전 통계학에서 표본은 모집단의 일부이며, 관찰을 통해 획득한 데이터로 그 개념을 제한하며, 반면에 모집단은 정보를 얻고자 하는 목표 대상의 전체 집단으로 바라본다. 한편, 모집단은 전수조사를 통하여만 파악 가능하고, 이러한 조사 과정은 매우 비효율적이다. 반면에 표본은 일부의 정보만 수집한 것이나, 적절한 통계적 도구를 활용하면 모집단의 형태를 구체화할 수 있다.

- 모집단Population
  - 정보를 얻고자 하는 목표 대상의 전체 집단
- 표본Sample

- 모집단의 일부이며, 관찰하여 획득한 데이터

- 모집단과 표본의 관계
  - 모집단은 전수조사를 통하여 파악 가능 ⇨ 비효율
  - 표본을 통하여 모집단의 정보를 추정

● 표본의 원천source

다양한 처리 과정을 거쳐 구체화 된 모집단의 의미를 파악하기 위하여, 다양한 곳으로부터 표본을 수집하는 것이 필수이다. 정부, 공공기관 등으로부터 공개된 데이터를 수집하는 것이 가능하고, 직접 실험을 통하여 획득하는 것 또한 가능하다. 사람을 대상으로 설문조사(survey)를 수행하여 데이터를 획득하는 것도 가능하며, 사물, 객체, 현상 등을 관찰하고 기록하여 데이터를 획득하는 것 또한 가능하다. 이처럼 다양한 표본의 원천이 있을 수 있으며, 그로부터 얻어낼 수 있는 다양한 모집단의 형태 또한 모델링 가능하다.

- 정부, 공공기관 등의 데이터 수집
- 실험을 통한 데이터 획득
- 설문조사 등을 통한 데이터 획득
- 사물, 객체, 현상의 관찰을 통한 데이터 획득

● 데이터의 유형

데이터는 크게 범주 데이터와 수치 데이터로 구분할 수 있다. 수치 데이터는 측정하여 숫자의 형태로 획득한 데이터이다. 예를 들어, 키, 몸무게, 온습도, 물품의 가격은 그 본질을 측정하여 측정 도구의 눈금치로 환산하여 표현한 것이다. 반면에 범주 데이터는 데이터의 범주를 사전에 정의하고, 데이터의 특성을 그 정해진 범주에 따라 분류한 것이다. 남녀의 성별, 직업, 시/군/구 등 거주구역별로 구분하는 것이 그러한 한 예라고 볼 수 있다.

- 수치 데이터Numerical Data
  - 측정하여 숫자의 형태로 획득한 데이터
  - 예) 키, 몸무게, 온습도, 물품의 가격
- 범주 데이터Categorical Data
  - 데이터의 범주(category)를 사전에 정의하고, 데이터의 특성을 범주에 따라 분류한 데이터
  - 예) 남녀 성별, 직업, 시/군/구 등 거주구역

● 통계의 분류

통계는 기술통계와 추측통계로 구분하는 것이 가능하다. 기술통계는 표본을 수집하여 정리 및 요약하는 과정으로, 수집한 데이터로부터 의미있는 정보를 추출하는 것을 그 목적으로 한다. 반면 추측통계는 표본을 분석하여 모집단의 정보를 추측하는 과정으로, 모집단의 정보 추측의 품질을 높이는 것을 그 목적으로 한다.

- 기술통계Descriptive Statistics
  - 표본을 수집 ⇨ 정리 및 요약
  - 목적 : 수집한 데이터로부터 의미있는 정보를 추출
- 추측통계Inferential Statistics
  - 표본을 분석 ⇨ 모집단의 정보를 추측
  - 목적 : 모집단의 정보 추측의 품질을 높임

● 통계 자료의 요약

통계 자료를 요약하는 고전적인 방법은 다양하게 존재한다. 여기서는 도수분포표와 히스토그램을 알아보도록 하자.

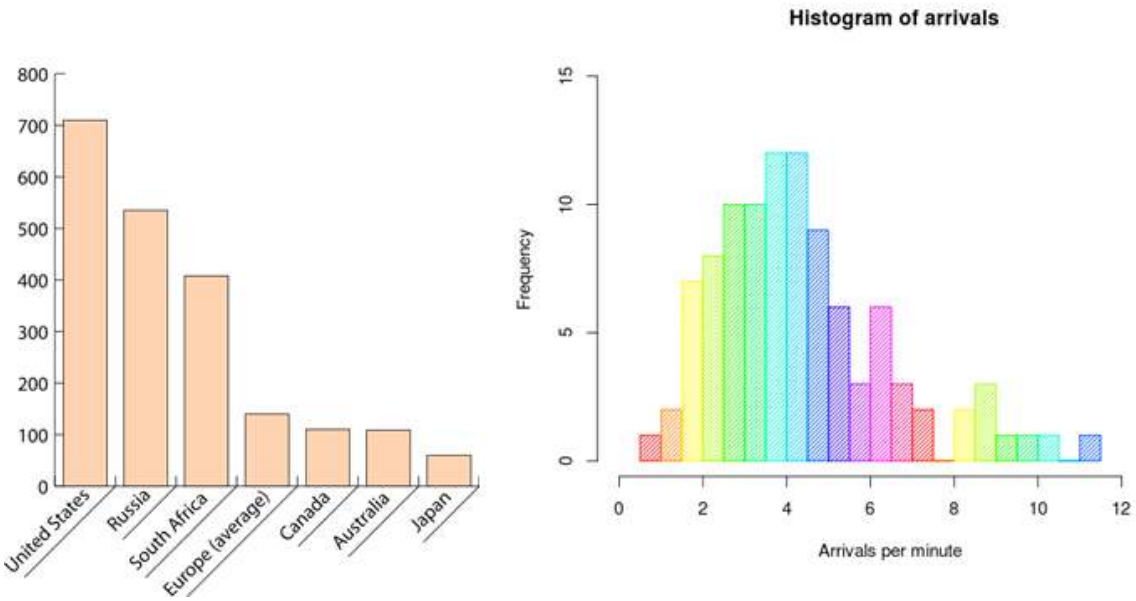
도수분포표는 구간/범주를 나누고 각 구간/범주별로 표본의 출현 빈도수를 표현한 표이다. 구체적인 수치를 통한 분석이 가능하다는 특성을 가지고 있다. 반면, 히스토그램은 구간/범주별 빈도수를 그림과 같은 시각화 표현을 한 것이다. 특히 양(magnitude)을 직관적으로 표현한다는 장점이 있다.

- 도수분포표Frequency Distribution Table
  - 구간/범주별로 표본의 출현 빈도수를 표현한 표
  - 구체적인 수치를 통한 분석 가능
- 히스토그램Histogram
  - 구간/범주별 빈도수를 그림으로 시각화 표현
  - 양(magnitude)을 직관적으로 표현

- 도수분포표의 예시

점수	의미	빈도수
5	매우 동의한다	20
4	동의한다	30
3	확실하지 않다	20
2	동의하지 않는다	15
1	매우 동의하지 않는다	15

- 히스토그램의 예시



● 통계 자료의 분석

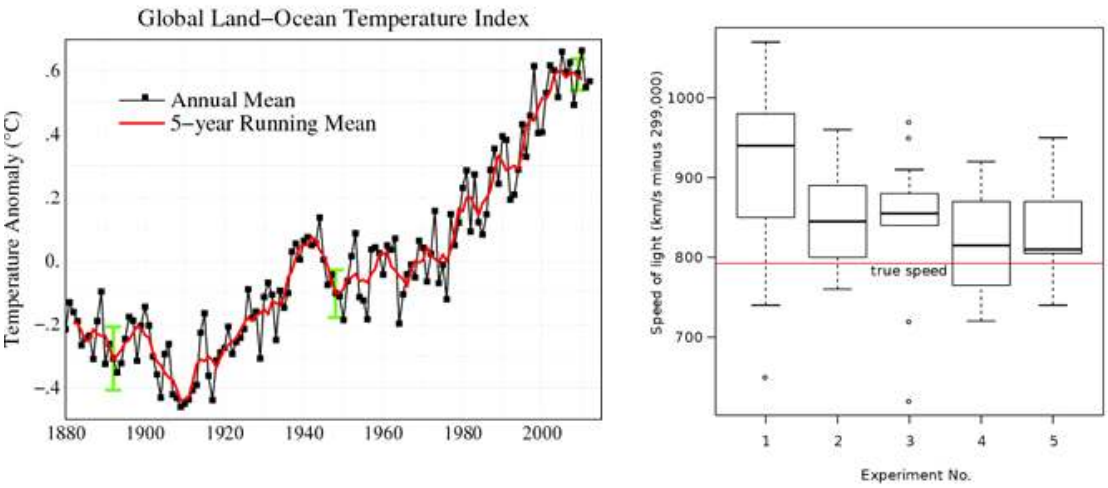
통계 자료를 분석하기 위한 고전적인 통계 도구는 다양하다. 여기서는 몇 가지만 소개해보도록 한다.

산술평균은 표본의 합을 표본의 수로 나눈 값으로, 우리가 일반적으로 “평균”이라고 일컫는 그것이다. 한편, 중앙값은 표본을 크기 순서로 나열하고, 중앙에 출현하는 값을 뜻한다. 최빈값은 표본 중 가장 큰 빈도수로 출현하는 값으로, 이 값을 이용하여 가장 자주 출현하는 경향성을 볼 수 있다. 범위는 표본의 가장 큰 값과 가장 작은 값 간의 차이를 뜻하며, 데이터가 존재하는 범위를 나타내는 수치이다. 마지막으로 표준편차는 분산을 구하여 제곱근을

취한 값으로, 표본이 얼마나 넓게 분포되어 있는지 나타내는 수치이다.

- 산술평균Mean/Average
  - 표본의 합을 표본의 수로 나눈 값
- 중앙값Median
  - 표본을 크기의 순서로 나열하였을 때 중앙에 출현하는 값
- 최빈값Mode
  - 표본 중 가장 큰 빈도수로 출현하는 값
- 범위Range
  - 표본의 가장 큰 값과 가장 작은 값의 차이
- 표준편차Standard Deviation
  - 분산의 제곱근으로 표본의 분포를 나타냄
- 분산Variance
  - 산술평균과 표본 간의 차이의 제곱합을 표표본의 수로 나눈 값

- 통계 자료 분석의 예시



● 확률 (Probability)

다음으로는 확률론에 대하여 학습하여 본다. 확률은 빅데이터를 바라보기 위한 강력한 도구의 하나이다. 확률은 모집단의 정보가 있는 경우 표현 가능하며, 모집단 대비 특정 사건이 발생하는 비율을 표현하는 것이다. 예를 들어 주사위를 무한 번 던져 3이 나오는 비율을 표현하는 것이 바로 확률이라 볼 수 있다. 우리는 주사위를 무한 번 던지면 3이 $\frac{1}{6}$의 비율로 나온다는 사실을 이미 알고 있다. 확률은 이처럼 사건을 일반화하고 사건의 경향을 알 수

있어, 미래에 대비할 수 있는 가능성을 열어주는 중요한 수학적 도구이다.

- 확률 Probability
  - 모집단의 정보가 있음
  - 모집단 대비 특정 사건이 발생하는 비율의 표현
  - 예) 주사위를 무한 번 던져 3이 나오는 비율의 표현
- 확률의 필요성
  - 사건의 일반화
  - 사건의 경향을 알 수 있음 ⇨ 미래에 대비 가능

● 확률의 유형

확률은 여기서 소개하는 것 외에도 다양한 유형이 있으나, 고전 확률론에서는 크게 세 가지의 확률 유형인 단순 확률, 결합 확률, 그리고 조건부 확률을 생각해볼 수 있다. 단순 확률은 한 가지의 사건이 발생할 확률을 뜻한다. 반면, 결합 확률은 두 가지 이상의 사건이 발생할 확률을 뜻한다. 조건부 확률은 특정한 사건이 발생하였다는 전제 하에 또다른 사건이 발생할 확률을 뜻한다.

- 단순 확률Simple Probability
  - 한 가지의 사건이 발생할 확률
- 결합 확률Joint Probability
  - 두 가지 이상의 사건이 발생할 확률
- 조건부 확률Conditional Probability
  - 특정 사건이 발생하였다는 전제 하에 또다른 사건이 발생할 확률

● 통계와 확률의 차이점

앞서 배운 통계와 확률에는 중요한 차이점이 있다. 통계는 모집단의 정보가 없으므로, 표본을 수집하여 모집단을 추정하고자 하는 목적을 가진다. 예를 들어, 주사위의 현상을 규명하기 위하여, 주사위를 실제로 100번 던져 3이 나오는 비율을 측정하는 것이 바로 통계적 접근 방법이다. 반면에, 확률은 모집단의 정보가 이미 있으며, 모집단에 대비하여 특정한 사건이 발생하는 비율을 표현하는 것이다. 예를 들어, 주사위를 실제로 무한 번 던지지는 않지만, 주사위를 무한 번 던진다고 가정할 때 3이 나오는 비율을 표현할 수 있는 것이 바로 확률이다.

- 통계
  - 모집단의 정보가 없음 ⇨ 표본으로 모집단 추정
  - 예) 주사위를 100번 던져 3이 나오는 비율을 측정
- 확률
  - 모집단의 정보가 있음
  - 모집단 대비 특정 사건이 발생하는 비율의 표현
  - 예) 주사위를 무한 번 던져 3이 나오는 비율의 표현

2. 확률 분포

우리는 앞서 통계와 확률에 대하여 좀더 엄밀한 정의를 내릴 수 있었다. 이러한 부분은 빅데이터를 조망하는 데에 있어서 큰 도움이 될 것이다. 다음 단계로, 확률 변수에 대하여 정의하고, 다양한 확률 분포에 대하여 조망해보도록 한다.

● 확률 변수 (Random Variable)

확률 변수는 2개 이상의 값을 취할 수 있는 변수이며, 통상 다루는 수학적 변수와는 다르게 취급된다. 확률 변수는 크게 나누어 이산확률변수와 연속확률변수로 나눌 수 있다. 이산확률변수는 값이 범주화 되어 있어 값을 나누어서 취급할 수 있는 변수이며, 반면에 연속확률변수는 값이 연속적이어서 값을 나누어 취급하지 않고 그 양 그대로 취급해야 하는 변수를 뜻한다.

- 정의
  - 2개 이상의 값을 취할 수 있는 변수
- 종류
  - 이산확률변수 : 값이 범주화 되어 있는 경우
  - 연속확률변수 : 값이 연속적인 경우

● 확률 분포

확률 분포는 데이터가 출현할 확률의 분포로 정의한다. 확률 분포는 데이터 출현 정도를 일반화함으로써 미래 예측을 가능케 하며, 표본 내에서 확률 변수의 출현 확률을 바탕으로 모집단에서 확률 변수를 추정 가능케 해준다.

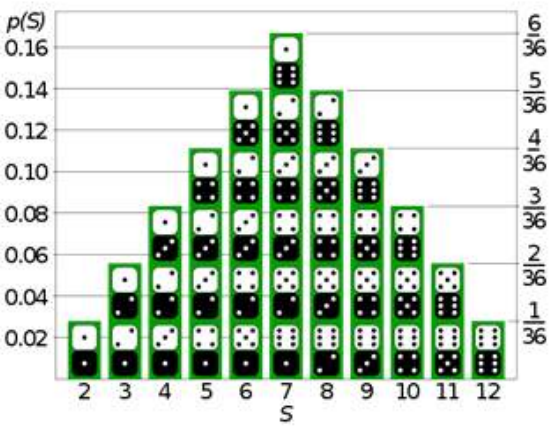
- 정의
  - 데이터가 출현할 확률의 분포
- 필요성
  - 데이터 출현의 정도를 일반화 ⇨ 미래의 예측 가능
  - 표본 내에서 확률 변수의 출현 확률을 바탕으로 모집단에서의 확률 변수의 출현 확률 추정 가능

● 확률 분포의 종류

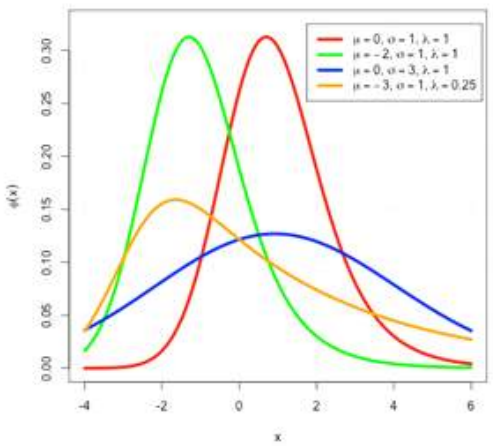
확률 분포는 크게 이산 확률 분포와 연속 확률 분포로 나눌 수 있다. 이산 확률 분포는 이산 확률 변수가 가지는 확률 분포이며, 수학적으로는 확률 질량 함수의 형태로 표현한다. 반면, 연속 확률 분포는 연속 확률 변수가 가지는 확률 분포로, 수학적으로는 확률 밀도 함수의 형태로 표현한다.

- 이산 확률 분포Discrete Probability Distribution
  - 이산 확률 변수가 가지는 확률 분포
  - 확률 질량 함수(Prob. Mass Function) 표현
- 연속 확률 분포Continuous Probability Distribution
  - 연속 확률 변수가 가지는 확률 분포
  - 확률 밀도 함수(Prob. Density Function) 표현

- 확률 분포의 예시



주사위 2개의 이산 확률 분포



EMG 분포

다음으로는 확률과 관련한 몇 가지 더 확장가능한 개념에 대하여 학습하여 보도록 하자.
기댓값은 확률 데이터가 집중되는 경향성을 대표하는 값으로, 통계적 관점에서는 평균이라

고도 한다. 한편, 분산은 확률변수가 기댓값으로부터 벗어난 정도를 표현하는 값으로, 그 자체만으로는 너무 값이 크기 때문에 큰 의미가 없다. 분산에 제곱근을 취하면 표준편차를 구할 수 있는데, 이는 기댓값 대비 분포의 정도를 효과적으로 표현할 수 있다. 따라서 빅데이터 분석을 할 때에는 반드시 기댓값과 표준편차는 반드시 산출하여야 한다.

- 기댓값Expectation
  - 확률 데이터가 집중되는 경향성을 대표하는 값
- 분산Variance
  - 확률변수가 기댓값으로부터 벗어난 정도를 표현
- 표준편차Standard Deviation
  - 분산의 제곱근으로, 기댓값 대비 분포 정도 표현

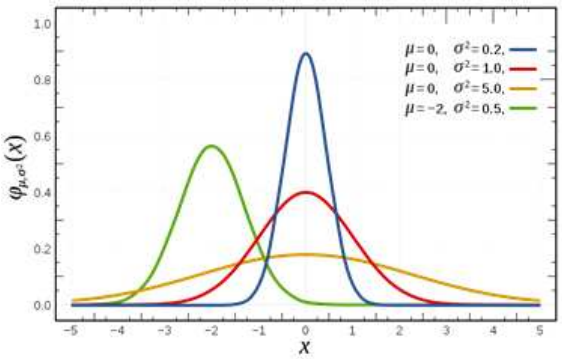
● 정규 분포Normal Distribution

다양한 확률 분포 중에서 정규 분포에 대한 이해는 확률 분포에 대한 이해에 있어 필수라고 볼 수 있다.

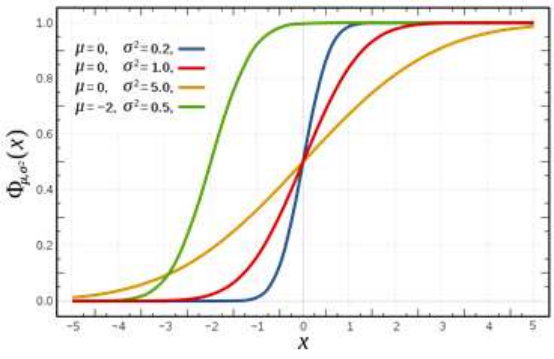
정규 분포는 평균과 표준편차 기반의 연속 확률 분포의 일종으로, 중심극한정리에 근거하고 있다. 중심극한정리란, 확률 변수의 평균은 항상 정규 분포에 근접하는 성질을 가지고 있는 것이다. 이에 근거하여, 우리는 대부분의 확률 변수의 평균을 정규 분포를 이용하여 분석할 수 있다.

- 정의
  - 평균과 표준편차 기반의 연속 확률 분포
  - 중심극한정리에 근거⇨ 확률 변수의 평균은 정규 분포에 근접하는 성질
- 특성
  - 절대근사한다.
  - 평균과 표준편차가 주어지면 ⇨ 엔트로피를 최대화
  - 정규 분포 곡선은 평균 대비 좌우 대칭
  - 중앙값의 확률이 최대

- 정규 분포의 예시



정규 분포의
확률 밀도 함수(PDF)



정규 분포의
누적 밀도 함수(CDF)

3. 모집단의 추정

우리는 앞서 통계와 확률의 고전 이론을 바탕으로, 통계와 확률의 큰 목적 중 하나가 바로 제한된 표본(=빅데이터)으로부터 모집단을 추정하는 것임을 학습하였다. 본 장에서는 모집단을 추정하기 위한 다양한 이론과 방법론에 대하여 학습하고, 특히 통계적 접근 방법에 대하여 정리하여 보도록 한다.

● 추정 이론

추정 이론은 통계학과 신호처리의 한 분야로, 표본을 바탕으로 인자를 추정하는 학문이다. 표본 이론은 한정된 데이터만 이용하여 최적의 추정 방법론(즉, 추정량)을 적용하는 것을 가능케 해준다.

- 정의
  - 통계학과 신호처리의 한 분야로, 표본을 바탕으로 인자(parameter)를 추정하는 학문
- 필요성
  - 한정된 데이터(=표본)를 바탕으로 최적의 추정 방법론(=추정량)을 적용 가능

● 추정 방법론 / 추정량 Estimation Methodology / Estimator

빅데이터에서 추정 방법론(추정량)은 지금 이 순간에도 다양한 방법들이 개발되고 있다. 여기서는 몇 가지 널리 알려진 추정 방법론들에 대하여 정리하고 가도록 하겠다.

- MLE : Maximum Likelihood Estimation
  - 사전 정보가 없는 상황에서 성능을 최대화하는 인자 추정 방법
- MAP : Maximum A Posteriori
  - 사전 정보나 그 가정을 바탕으로 성능을 최대화하는 인자 추정 방법
- 최소제곱법Least Squares
  - 사전 정보의 오차 제곱을 최소화하는 인자 추정 방법
- MMSE : Minimum Mean Squared Error
  - 사전 정보의 평균 제곱근 오차(MSE)를 최소화하는 인자 추정 방법
- 칼만 필터Kalman Filter
  - 이상 데이터가 포함된 선형 모집단의 인자를 추정하는 방법

● 모집단의 추정Estimating Population

표본으로부터 다양한 정보를 추출하고, 위에서 언급한 다양한 추정 방법론을 적용하여 모집단의 정보를 비교적 정확하게 추정하는 것이 가능하다. 특히, 대부분의 빅데이터 표본이 중심극한정리에 의하여 정규 분포에 가깝게 근사한다는 가설을 바탕으로 할 때, 우리는 빅데이터의 평균 및 비율 등을 구하는 것이 큰 목표 중 하나라고 볼 수 있다. 이러한 과정을 가리켜서 모집단의 추정이라고 하며, 이는 추정 이론의 한 갈래이다.

모집단을 전수조사하는 것은 인적 비용, 물적 비용 등으로 말미암아 분석으로 소요하는 비용이 분석으로부터 얻는 이득보다 더 커질 수 있다. 즉, 분석의 경제성이 하락할 수 있다. 따라서 올바른 추정 방법론을 적용하여 표본을 바탕으로 모집단을 정확하게 추정할 수 있는 경우, 분석의 비용을 절약하는 한편, 분석으로부터 얻는 이득을 극대화함으로써, 분석의 효율성을 향상시키는 것이 가능하다.

- 정의
  - 표본의 정보를 바탕으로 추정 방법론을 적용하여 모집단의 정보(평균, 비율)를 정확하게 추정
  - 추정 이론(Estimation Theory)의 한 갈래
- 필요성
  - 모집단을 전수조사하는 경우 ⇨ 분석 경제성 하락↓
  - 표본을 바탕으로 모집단을 정확하게 추정하는 경우⇨ 분석 비용 절약 가능, 분석 효율성 향상

● 분산의 종류

분산은 크게 모 분산과 표본 분산으로 나눌 수 있다. 모 분산은 모집단으로부터 구한 분산이며, 표본 분산은 표본으로부터 구한 분산이다.

- 모 분산Population Variance
  - 모집단으로부터 구한 분산
- 표본 분산Sample Variance
  - 표본으로부터 구한 분산

● 표준편차의 종류

한편, 표준편차 또한 모 표준편차와 표본 표준편차로 구분할 수 있다. 모 표준편차는 모집단으로부터 구한 표준편차이며, 표본 표준편차는 표본으로부터 구한 표준편차이다.

- 모 표준편차Population Std. Dev.
  - 모집단으로부터 구한 표준편차
- 표본 표준편차Sample Std. Dev.
  - 표본으로부터 구한 표준편차

● 모집단 평균 추정Estimating Population Mean

모집단의 평균은 표본의 평균, 표본의 수, 표본의 표준편차, 그리고 신뢰구간 상수를 통하여 구할 수 있다.

- 신뢰구간 추정

$$\bar{x} - Z_{\alpha/2}\frac{s}{\sqrt{n}} \leq \mu \leq \bar{x} + Z_{\alpha/2}\frac{s}{\sqrt{n}}$$

● 모집단 비율 추정Estimating Population Ratio

모집단의 비율은 표본의 평균비율, 표본의 수, 그리고 신뢰구간 상수를 통하여 구할 수 있

다.

- 신뢰구간 추정

$$\bar{p} - Z_{\alpha/2}\sqrt{\frac{\bar{p}(1-\bar{p})}{n}} \leq p \leq \bar{p} + Z_{\alpha/2}\sqrt{\frac{\bar{p}(1-\bar{p})}{n}}$$

8. 데이터 통계 분석 (2)

1. 분산분석과 상관분석

빅데이터에서는 두 개 이상의 다수의 분포를 비교하여 그로부터 다양한 의미를 추출하는 것이 중요할 수 있다. 이에, 본 장에서는 두 개 이상의 다수의 분포를 분석하기 위한 도구로, 분산분석과 상관분석에 대하여 학습하는 것을 목표로 한다.

● 분산분석 (ANOVA)

분산분석은 통계학에서 두 개 이상의 다수의 집단을 비교할 때, F분포를 이용하여 가설의 검정을 수행하는 방법론이다. 이 때, 분산분석의 영어 약어인 ANOVA는 분산의 분석(analysis of variance)를 직역한 것이다. 분산분석은 통계학자 로날드 피셔(R.A. Fisher)에 의하여 1920년대부터 1930년대에 걸쳐 작성되었다.

- 정의
  - 통계학에서 두 개 이상의 다수의 집단을 비교할 때 F분포를 이용하여 가설검정을 하는 방법
  - ANOVA = ANalysis Of VAriance
- 역사
  - 통계학자 로날드 피셔 (R.A. Fisher)에 의해 1920년대 ~ 1930년대에 걸쳐 작성

● F분포

F분포는 분산분석에서 사용하는 중요한 측정 도구 중 하나로, 분산의 비교를 통하여 얻어지는 분포 비율로 정의된다. F분포는 군 사이의 변동을 군 내부의 변동으로 나눔으로써 표현할 수 있다.

F분포는 우선 집단 간에 어떠한 동질성이 있다는 것을 가정하고 분석한다. 따라서 군내변동이 클 경우 집단 간의 평균 차이를 확인하는 것이 쉽지 않다. 이로 인하여 분산분석의 결과, 분산 차이가 큰 경우에는 그러한 차이를 발생시키는 원인을 제거하는 것이 필요하다.

- 정의
  - 분산의 비교를 통하여 얻어지는 분포 비율
  - F = (군간변동) / (군내변동)

- 특성
  - 집단 간의 동질성을 가정하고 분석
  - 군내변동이 크면 → 집단 간 평균차이 확인 어려움
  - 분산 차이가 큰 경우 → 유발 원인 제거 필요

● F분포의 가정

F분포는 정규성, 분산의 동질성, 관찰의 동질성을 가정하고 있다.

- 1. 정규성 가정
  - 모집단에서 변인 Y는 정규분포를 따른다.
  - 모집단에서 변인 Y의 평균은 다를 수 있다.
- 2. 분산의 동질성 가정
  - Y의 모집단 분산은 각 모집단에서 동일하다.
- 3. 관찰의 독립성 가정
  - 각 모집단에서 크기가 서로 다른 표본이 독립적으로 표집된다.

● F분포의 계산

F분포는 모집단 분산의 추정피 비율을 계산함으로써 알아낼 수 있다. 여기서 F값은 이론적 확률분포의 하나인 F분포를 따르게 된다.

- 모집단 분산의 추정치 비율을 계산
- F값은 이론적 확률분포인 F분포를 따름

● 분산분석의 모형

분산분석은 고정효과 모형, 무선효과 모형, 혼합효과 모형 등을 가진다. 각각의 특징은 다음과 같다.

- 고정효과 모형
  - 수준의 선택이 기술적으로 정해지고 각 수준이 기술적 의미를 가진 효과 인자

- 무선효과 모형
  - 수준의 선택이 임의로 이루어지며 각 수준이 기술적 의미를 가지지 않은 효과 인자
- 혼합효과 모형
  - 고정효과 인자와 무선효과 인자가 함께 사용된 경우

● 분산분석의 종류

분산분석은 단순히 F의 값을 이용하는 것 외에, 종속변인과 독립변인의 형태에 따라 다양한 종류를 가진다. 이 중 몇 가지를 소개하면 다음과 같다:

- 일원분산분석one-way ANOVA
  - 종속변인이 1개이며 독립변인 집단도 1개
- 다원변량분산분석MANOVA
  - 독립변인의 수가 2개 이상일 때 집단 비교
- 공분산분석ANCOVA
  - 두 개 이상 종속변인이 관계된 상황에 적용
- 이원분산분석two-way ANOVA
  - 특정한 독립변인 위주로 분석하고 다른 독립변인은 통제변수로 설정 분석

● 상관분석

두 분포를 비교하는 또 다른 큰 줄기의 하나인 상관분석은 빅데이터 분석에 있어 널리 활용되고 있다. 특히 미지의 두 변수 간의 상관성을 알기 위해서는 굳이 선형이 아니더라도 우선 상관분석을 행하여 보는 것이 바람직하다. 이렇게 제1의 방법으로 택하여지는 상관분석에 대하여 학습하여 보도록 하자.

상관분석은 두 변수 간의 선형적 관계를 분석하는 방법으로, 두 변수가 독립적이거나 혹은 상관적임을 가정하고, 두 변수 간의 강도를 측정하는 것을 목표로 한다. 이 때, 측정된 두 변수 간의 강도를 상관관계(correlation)라고 한다.

- 정의
  - 두 변수 간의 선형적 관계를 분석하는 방법
  - 두 변수는 독립적이거나 상관될 수 있다.

- 두 변수 간의 강도를 상관관계라고 함(Correlation, Correlation coefficient)

● 상관분석의 가정

상관분석은 선형성, 정규분포성, 무선독립표본, 그리고 동변량성을 가정하고 있다. 각 가정의 특성을 정리하면 다음과 같다.

- 선형성
  - 두 변인 X, Y의 직선적인 정도, 선점도를 사용
- 정규분포성
  - X의 값에 관계없이 Y의 흩어진 정도가 같은 것이 분산성의 반대어
- 무선독립표본
  - 두 변인의 측정치 분포는 모집단에서 정규분포
- 동변량성
  - 모집단에서 표본을 추출할 때 표본대상이 확률적으로 선정되는 것

● 상관분석의 분석방법 ①

상관분석의 분석 방법은 다양하게 제안되어 있으며, 여기서는 몇 가지만 소개해보도록 한다.

- 1. 피어슨 상관계수Pearson Correlation Coefficient
  - 두 변수 간의 관련성을 구하기 위하여 보편적으로 이용
- 2. 스피어만 상관 계수Spearman Correlation Coefficient
  - 데이터가 서열척도(순위값)인 경우의 상관계수
  - 데이터를 작은 것부터 차례로 순위를 매겨 서열 순서로 바꾼 뒤 순위를 이용하여 상관계수 계산
  - 자료에 이상점이 있거나 표본크기가 작을 때 유용
  - 상관계수값에 따른 분류
  - +1 = 두 변수 안의 순위가 완전히 일치
  - -1 = 두 변수 안의 순위가 역순인 경우
- 3. 크론바흐 알파 계수 신뢰도Cronbach's Alpha
  - 검사의 내적 일관성을 나타내는 값을 계산

- 한 검사 내에서 변수들 간의 평균상관관계에 근거하여 검사문항들이 동질적 요소로 구성되어 있는지를 분석하는 방법
- 동일한 경우에는 결과가 비슷하며, 동일하지 않은 경우에는 결과가 상이

2. 회귀분석

빅데이터의 분석 도구 중 회귀분석은 모델링을 가정하고 이에 따라 주어진 변수 간의 모델을 형성하는 기법이다. 회귀분석에 대하여 학습함으로써 우리는 변수를 모델링하는 또 하나의 강력한 가설 기반 방법론을 얻을 수 있다.

● 회귀분석 (regression analysis)

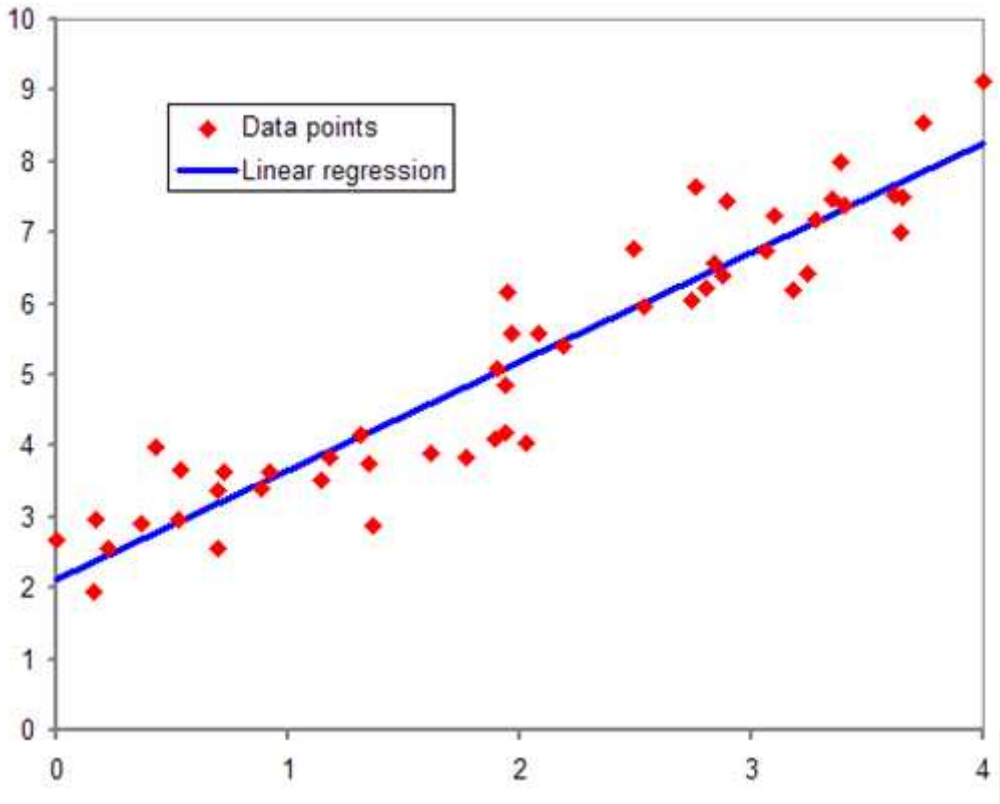
회귀분석은 관찰된 연속형 변수들 간의 모형을 구한 뒤 적합도를 측정하는 분석 방법이다.

- 정의
  - 관찰된 연속형 변수들 간의 모형을 구한 뒤 적합도를 측정하는 분석 방법

● 회귀분석의 가정

- 오차항은 모든 독립변수에 대하여 동일 분산을 가짐
- 오차항의 평균(기댓값)은 0이다.
- 수집된 데이터의 확률 분포는 정규분포를 이룬다.
- 독립변수 간에는 상관관계가 없어야 한다.
- 시간에 따라 수집된 데이터는 잡음 영향이 없다.

● 회귀분석의 예시



3. 시계열 데이터의 분석

최근, 빅데이터로 수집하는 데이터의 형태 중 시계열 데이터의 중요도가 높아지고 있다. 시계열 데이터는 시간의 정기적 혹은 부정기적 흐름에 따라 수집된 데이터이다. 데이터를 수집한 기간이 축적될수록 과거로부터 현재, 미래의 다양한 추세를 알 수 있어, 미래를 예측하는 데에 아주 큰 도움을 준다. 본 장에서는 시계열 데이터를 분석하기 위한 일반적인 방법론에 대하여 학습하여 보도록 한다.

● 시계열 데이터

시계열 데이터란, 일정 간격으로 배치된 데이터의 수열로, 반드시 시간을 기반으로 할 필요는 없다. 그러나, 일반적으로 인과성을 가지고 데이터가 축적되므로, 사실상 데이터가 쌓이는 순서는 시간에 따른 순서인 경우가 많다.

시계열 해석이란, 시계열을 해석하고 이해하는 방법으로, 특히 시계열 데이터가 어떠한 법칙을 통하여 생성되는지 밝혀내는 방법을 뜻한다.

시계열 예측이란, 시계열 데이터를 기반으로 수학적 모델을 구축하고, 미래에 발생하는 시계열의 형태를 예측하는 작업이다. 공학, 과학, 금융시장 등에서 널리 활용되고 있다.

- 시계열, Time Series
  - 일정 간격으로 배치된 데이터의 수열
- 시계열 해석, Time Series Analysis
  - 시계열을 해석하고 이해하는 방법
  - 시계열이 어떠한 법칙에서 생성되는지 밝혀내는 작업
- 시계열 예측Time Series Prediction
  - 시계열을 기반으로 수학적 모델을 구축하고 미래에 발생하는 시계열의 형태를 예측하는 작업
  - 공학, 과학, 금융시장 등에서 사용

● 시계열 데이터 분석

시계열 데이터를 분석하는 방법에는 다양한 방법이 존재한다. 여기서는 AR, MA, I 모델 등을 소개하도록 한다.

- Autoregressive (AR) 모델
- Moving Average (MA) 모델
- Integrated (I) 모델

● Autoregressive (AR) 모델

AR 모델은 그 이름에서부터 알 수 있듯, 자기상관성 정보에 기반한 모델이다. 보다 구체적으로 이야기하자면, 어떠한 변인에 의하여 이전 값이 이후 값에 미치는 영향을 모델링하는 기법이다. 예를 들어 이전의 값이 감소하는 경우에는 이후의 값이 감소하는 형태를 모델링하여 그 추세를 찾아내는 것이 AR 모델의 특성이다.

- 자기상관성(autocorrelation) 정보에 기반
- 어떠한 변인에 대하여 이전의 값이 이후의 값에 미치는 영향을 모델링
- 예) 이전의 값이 감소하면 이후의 값 또한 감소

● Moving Average (MA) 모델

MA 모델은 어떤 변수의 평균값이 지속적으로 증가하거나 감소하는 경향에 대한 모델링이다. 예를 들어 봄에서 여름이 되면 전기 수요가 대체로 증가하는 것을 모델링하는 것이 바로 MA 모델의 접근 방법이라 볼 수 있다.

- 어떤 변수의 평균값이 지속적으로 증가하거나 감소하는 경향에 대한 모델링

- 예) 봄에서 여름이 되면 전기 수요가 대체로 증가

● ARMA 모델

ARMA 모델은 AR 모델과 MA 모델을 결합한 형태의 모델이다. 즉, 이전의 값으로부터 이후의 값을 모델링할 수 있는 한편, 일반적인 추세 또한 동시에 모델링할 수 있음으로써, 다양한 통합 모델을 도출할 수 있다는 장점을 가지고 있다.

- 기존의 AR 모델과 MA 모델을 통합하여 다양한 통합 모델이 도출될 수 있음

● Integrated (I) 모델

Integrated 모델은 AR과 MA 모델 외에도 추세 등을 반영하여 다양한 통합 모델을 도출해 낼 수 있다. 특히 ARIMA 모델은 시계열의 분석(모델링) 방법 중 상당히 대중적으로 성공한 모델이다.

- AR 모델, MA 모델 외에 통합 모델을 고려
- 과거의 데이터 뿐만 아니라 추세(momentum)까지 반영한 모델

9강. 데이터 마이닝

1. 데이터 마이닝의 정의와 이해

빅데이터의 분석 방법론을 이해하는 데에 있어서 데이터 마이닝의 일반적인 접근 방법과 구체적인 방법론을 이해하는 것은 중요하다. 특히, 지금까지 배운 R과 같은 빅데이터 처리 및 분석 도구를 기반으로, 데이터 마이닝의 방법론을 적용하면 수집한 빅데이터로부터 보다 다양한 결론을 도출해낼 수 있고 부가가치를 이끌어낼 수 있다. 본 장에서는 데이터 마이닝에 대하여 정의하고, 일반적인 접근 방법에 대하여 이해해보도록 한다.

● 데이터 마이닝 Data Mining

데이터 마이닝이란, 대규모로 저장된 데이터 안에서 체계적이고 자동적인 통계적 규칙이나 패턴을 찾아내는 일련의 작업을 뜻한다. 특히, 인간의 시선으로 수집된 빅데이터를 한 번에 조망하고 그것으로부터 인사이트(insight)를 이끌어내는 것이 거의 불가능해짐에 따라, 데이터 마이닝을 기반으로 한 다양한 자동화 도구를 통하여 인사이트를 찾아내는 것이 필수가 되었다. 즉, 수습 불가능한 형태가 된 대규모 데이터로부터 의미를 찾아내는 데에 있어 데이터 마이닝의 중요도가 급상승하고 있다.

- 정의
  - 대규모로 저장된 데이터 안에서 체계적이고 자동적인 통계적 규칙이나 패턴을 찾아내는 작업
  - KDD (Knowledge-Discovery in Databases)
- 빅데이터 시대의 의미
  - 대규모 데이터로부터 의미를 찾아내는 데 있어 그 중요도가 급증

● 데이터 마이닝의 적용 분야

데이터 마이닝은 분류, 연관성 분석, 연속성 분석, 예측 분석, 군집화 분석 등 다양한 분야에 적용하는 방법론으로 구성되어 있다.

- 분류classification
  - 일정한 집단에 대한 특정 정의를 통하여 분류 및 구분의 형태를 추론하는 분야
- 연관성association
  - 동시에 발생한 사건 간의 관계를 정의하는 분야

- 연속성sequencing
  - 특정 기간에 걸쳐 발생하는 관계를 규명, 연관성 분석과 달리 기간 특성을 고려
- 예측forecasting
  - 빅데이터 집합 내의 패턴을 기반으로 미래에 발생하는 데이터의 형태를 예측하는 분야
- 군집화clustering
  - 구체적인 특성을 공유하는 군집(cluster)을 찾음
  - 미리 정의된 특성 정보가 없이 군집을 탐색

2. 데이터 마이닝 방법론

데이터 마이닝은 다음과 같은 일반적인 방법론을 가지며, 빅데이터의 전체 단계와도 겹치는 부분들이 많다. 이러한 부분에 대하여 학습하면 다음과 같다:

- **데이터 마이닝 방법론**
  - 1. 프로젝트의 목적과 적용 가능성을 확인한다.
  - 2. 분석에서 사용할 데이터를 수집한다.
  - 3. 데이터를 전처리한다.
  - 4. 데이터를 축소하고 분할한다.
  - 5. 데이터 마이닝 기법을 선택한다.
  - 6. 데이터 마이닝을 수행한다.

데이터 마이닝 방법론의 각 단계별 특성은 다음과 같다.

- **1. 프로젝트의 목적과 적용 가능성을 확인한다.**
  - 일회성 프로젝트인 경우→ 프로젝트의 목적을 수립한다.
  - 연속성 프로젝트인 경우,→ 프로젝트의 적용 가능성을 확인한다.
- **2. 분석에서 사용할 데이터를 수집한다.**

이 단계에서는 데이터베이스로부터 무작위 표본을 추출하거나, 내부 데이터와 외부 데이터를 수집함으로써 데이터 수집을 완성하는 단계이다. 수집 방법론으로는 수집 데이터를 선정하고, 세부 계획을 수립하고, 이후 테스트 수집을 진행한 후 수집을 진행하는 과정을 거친다.

- 개요

- 데이터베이스에서 무작위 표본을 추출
- 내부 데이터와 외부 데이터를 수집

- 수집 방법론

- 수집 데이터를 선정
- 세부계획을 수립
- 테스트 수집 진행 후 수집 진행

● 3. 데이터를 전처리한다.

데이터 마이닝에서 데이터의 전처리는 필수 과정이라 할 수 있다. 본 단계에서는 데이터의 조건을 검증하고 정제한다. 특히, 산점도, 행렬표 등 다양한 그래프 도구를 사용하여 분석하는 것을 포함한다. 변수에 대하여 명확히 정의하고, 측정단위나 측정기간 등에 대한 일관성의 확인 또한 필수이다. 본 과정에서는 결측치, 변수의 값의 범위, 극단치 등에 대한 고려가 필요하다.

- 개요

- 데이터의 조건을 검증하고 정제한다.
- 산점도, 행렬표 등 그래프를 사용하여 분석
- 변수에 대한 정의, 측정단위, 측정기간 등에 대한 일관성 확인

- 고려 사항

- 결측치를 어떻게 처리해야 하는가?
- 각 변수의 값이 합리적인 범위 내에 있는가?
- 극단치(최대/최소)가 존재하는가?

● 4. 데이터를 축소하고 분할한다.

큰 데이터가 항상 이후의 단계에 좋은 것은 아니다. 합리적인 크기로 데이터를 축소하고 분할함으로써 보다 효율적인 분석을 할 수도 있다. 특히 이 단계에서는 불필요한 변수를 제거하고, 분석가능한 형태로 변수의 형태를 전환하며, 새로운 변수를 생성하기도 하며, 데이터를 다양한 집합으로 분할하기도 한다. 특히 데이터를 학습용, 평가용, 검증용 데이터로 분류하는 것이 필요하다.

- 개요

- 불필요한 변수를 제거
- 변수를 분석가능한 형태로 변환
- 새로운 변수를 생성

- 데이터를 다양한 데이터 집합으로 분할

- 데이터 집합의 종류
  - 학습용 데이터 (training)
  - 평가용 데이터 (test, evaluation)
  - 검증용 데이터 (verification)

● 5. 데이터 마이닝 기법을 선택한다.

본 단계에서는 적절한 데이터 마이닝 기법을 선택함으로써 프로젝트에 적합한 분석 유형을 결정할 수 있다. 특히 고려해야 할 기법으로는, 고전 분석 모델, 딥러닝 분석 모델, 계층적 군집 분석 등이 있다.

- 개요
  - 프로젝트에 적합한 분석 유형을 결정
- 기법
  - 분산분석, 상관분석, 회귀분석 등 고전 분석 모델
  - 신경망 모형 등 딥러닝 분석 모델
  - 계층적 군집 분석 등

● 6. 데이터 마이닝을 수행한다.

이제 본격적으로 데이터 마이닝을 수행하는 단계에 이르렀다. 본 단계에서는 이전에 결정한 사항들을 기반으로 데이터 마이닝을 수행한다. 다양한 변인을 적용하여 분석을 수행하고, 평가용 데이터를 이용하여 수행 후 개선되는 변인을 토대로 적용한다. 마지막으로 수행 결과 구축된 모델을 바탕으로 목표로 하는 응용에 시험 적용하여 본다.

- 개요
  - 이전에 결정한 사항을 토대로 데이터 마이닝을 수행
  - 다양한 변인을 적용하여 분석 수행
  - 평가용 데이터를 이용하여 수행 후개선되는 변인을 토대로 적용
  - 수행 결과로 구축된 모델을 바탕으로 시험 적용

● 학습 방법론

데이터 마이닝에서는 다양한 방법론을 적용하게 된다. 특히 신경망 네트워크, 딥러닝 등에 적용하기 위한 학습 방법론은 그 숙지가 필수이다. 특히 지도학습, 자율학습, 반지도학습 등에 대한 구분은 필수라고 볼 수 있다.

- 지도학습Supervised Learning
  - 출력 데이터에 맞게 출력되도록 학습용 데이터셋을 이용하여 예측변수와 출력변수 간의 관계를 학습
- 자율학습Unsupervised Learning
  - 출력변수가 명확히 정의되지 않은 경우, 예측변수에 대한 자율학습을 통하여 모델을 구축하는 방법
- 반지도학습Semi-supervised Learning
  - 지도학습과 자율학습의 방식을 조합하여 예측변수와 출력변수 간의 일부를 자율적으로 모델링하는 한편, 정해진 데이터셋에 대한 학습 또한 수행

3. 데이터 마이닝 적용 사례

데이터 마이닝 방법론을 통하여 공공서비스, GPS 시스템, 보건/의료, 제조/물류/마케팅 등 다양한 분야에서 혁신을 이끌어낼 수 있다. 다음은 그 사례들에 대하여 열거한 것이다.

- **공공시스템**
  - 국세청의 탈세 방지 시스템에 적용
  - 사기방지 솔루션, 소셜 네트워크 분석, 지능형 감지 시스템 구축
  - 세금 누락 및 불필요한 세금 환급 절감 효과 발생
  - 탈세자 수 감소 및 범죄 사건 미연 방지 가능
- **GPS 시스템**
  - 자동차의 센서 데이터(예:GPS)를 통하여 교통 정보 수집
  - 지능형 교통 정보 시스템을 구축 가능
  - 실시간 교통 정보를 공유하여 최적의 교통 안내 서비스
  - 불필요한 에너지 낭비 방지 및 교통 시스템 효율 증대
- **보건/의료**
  - 유전자 정보를 토대로 질병 연구에 활용
  - 새로운 질병에 대한 빠른 진단 서비스
  - 난치병 및 불치병 관련 유전자 정보를 토대로 신치료제 개발
  - 최신 IT 기술 결합으로 치료 확률 상승

- **제조/물류/마케팅**
  - 소비자의 니즈를 예측하여 제품을 미리 제조 및 배급
  - 제조/물류/마케팅 비용을 최소화 가능
  - 제품의 소비자 도달 시간 최소화

10강. 정형 데이터 마이닝

1. 분류(Classification) 분석

수집한 빅데이터를 특정한 범주로 자동 분류할 수 있다면 분류의 효율성이 향상될 것이다. 정형 데이터 마이닝의 한 목표는 빅데이터로부터 적절한 모델링을 통하여 좋은 분류 모델을 구축하는 것이다. 본 장에서는 정형 데이터 마이닝에서 분류의 정의에 대하여 알아보고, 각 분류 방법론에 대하여 학습하여 보도록 한다.

● 분류 (Classification)

분류란, 데이터가 어느 그룹에 속하는 지 예측하는 데에 사용하는 데이터 기법이다. 분류는 군집화와 유사하게 데이터를 나눈다는 관점이 있으나, 군집화와 다르게 각 데이터의 범주(계급)이 어떻게 정의되는 지 알고 있어야 한다.

- 정의
  - 데이터가 어느 그룹에 속하는지 예측하는 데에 사용하는 데이터 기법
  - 군집화(clustering)와 유사하나,각 계급이 어떻게 정의되는지 미리 알아야 함

● 분류 방법론

분류 방법론으로는 의사결정나무, 베이지안 정리, 인공 신경망, 지지 벡터 기계 등이 있다. 각 방법론의 특성을 정리하면 다음과 같다:

- 의사결정나무Decision Tree
  - 어떤 항목에 대한 관측값과 목표값을 연결시키기 위한 트리 구조를 결정
- 베이지안 정리Bayesian Theorem
  - 불확실성 하에서 분류 문제를 조건부 확률의 방법으로 해결하는 방법
- 인공 신경망Artificial Neural Networks
  - 생물학의 신경망에서 영감을 얻은 방법론으로, 시냅스를 모델링하여 모델 구축
- 지지 벡터 기계Support Vector Machines
  - 주어진 데이터 집합을 바탕으로 새로운 데이터 소속 그룹을 판단하는 모델 구축

● 의사결정나무Decision Tree

의사결정나무는 결정 트리 학습법이라고도 한다. 의사결정나무는 관측값과 목표값 간의 모델을 구축하는 것을 최종 목표로 한다. 의사결정나무의 갈래로는 분류 트리 분석, 회귀 트리 분석 등이 존재한다.

의사결정나무는 주로 지도 분류 학습법에서 사용한다. 특히 분류 속도를 향상시키기 위하여 랜덤 포레스트 등의 방법을 적용할 수 있다.

- 개요
  - 결정 트리 학습법이라고도 하며, 관측값과 목표값 간의 모델을 구축
  - 분류 트리 분석, 회귀 트리 분석 등 존재
- 특징
  - 지도 분류 학습법에서 주요 사용
  - 랜덤 포레스트 (Random Forest) 등의 방법을 이용하여 분류 속도 향상 가능

● 베이지안 정리Bayesian Theorem

베이지안 정리는 이전에 학습한 조건부 확률 방법론에 기반한 모델링 방법이다. 특히 지도 학습 환경에서 효율적인 훈련이 가능한 특성이 있다.

베이지안 정리는 최대우도방법(MLE)을 이용하여 모수 추정을 수행한다. 베이지안 정리를 이용하여 도출된 모델은 많은 응용에서 복잡한 실제 상황에서 잘 작동함이 검증되어 있어, 지금도 널리 활용중에 있다. 특히, 불확실성 하에서 의사결정 문제를 확률론적으로 다룰 때 주로 사용하는 방법이다.

- 개요
  - 조건부 확률 모델에 기반
  - 지도 학습 환경에서 효율적 훈련 가능
- 특징
  - 최대우도방법(MLE)를 이용하여 모수 추정 수행
  - 복잡한 실제 상황에서 잘 작동함이 검증
  - 불확실성 하에서 의사결정 문제를 확률론적으로 다룰 때 사용하는 방법

● 인공 신경망Artificial Neural Networks

인공 신경망을 이용하여 기계학습과 인지과학에서 생물학의 신경망으로부터 영감을 얻은 통계학적 학습 알고리즘을 분류에 적용할 수 있다. 특히, 지도/반지도/자율 학습을 모두 적용하여 각기 다른 다양한 결론을 이끌어낼 수 있다.

인공 신경망은 역전파 기법과 기반경사 하강법을 기반으로 학습이 이루어진다. 그러나 인공 신경망은 태생적으로 항상 최적의 해를 찾아내지 못하는 문제가 있다. 이러한 문제를 해결하기 위하여 유전 알고리즘 등 다양한 방법을 적용함으로써 보다 최적의 해에 가까운 해를 찾아내는 등, 학습 효과를 극대화하는 것이 가능하다.

- 개요
  - 기계학습과 인지과학에서 생물학의 신경망으로부터 영감을 얻은 통계학적 학습 알고리즘을 분류에 적용
- 특징
  - 지도/반지도/자율 학습 모두 적용 가능
  - 역전파 기법(Backpropagation) 기반경사 하강법(Gradient Descent)
  - 유전 알고리즘 등 다양한 방법을 이용해 학습 효과 극대화 가능

● 지지 벡터 기계Support Vector Machines

지지 벡터 기계는 유한 차원 공간에서 데이터를 분류하기 위한 최적 초평면을 모델링하는 방법이다. 특히, 데이터의 수가 적을 때에도 그 일반화 성능이 뛰어나다고 알려져 있다.

지지 벡터 기계 중 선형 지지 벡터 기계는 표본에 대한 최적의 초평면 모델을 모델링하는 것이 가능하다. 특히 소프트 마진 기반의 뛰어난 일반화 성능 덕택에, 새롭게 발생하는 표본에 대하여 분류 성능을 극대화할 수 있다. 한편, 커널 트릭을 이용하여 비선형 데이터에 대한 분류 또한 가능할 수 있다.

- 개요
  - 유한 차원 공간에서 데이터를 분류하는 최적 초평면(hyperplane)을 모델링
- 특징
  - 선형 SVM을 이용하여 현존하는 표본에 대한 최적의 초평면 모델을 모델링 가능
  - 새롭게 발생하는 표본에 대하여 분류 성능을 극대화하기 위한 소프트 마진
  - 커널 트릭(kernel trick)을 이용하여 비선형 분류 또한 가능

2. 군집(Clustering) 분석

정형 데이터 마이닝 분석의 다른 한 갈래인 군집 분석은 데이터의 특성에 대한 정보가 부족하며, 데이터에 대한 명확한 범주가 정의되지 않았을 때 인사이트를 얻기 위한 방법론의 하나이다. 본 장에서는 군집 분석에 대하여 정리하고 학습하여 보도록 한다.

● 군집(Clustering)

군집은 데이터 마이닝 기술의 한 방법으로, 빅데이터에서 데이터의 특성을 고려하여 군집을 정의하고, 각 군집의 대표점을 찾는 방법이다. 특히 빅데이터 분석에 있어 데이터를 분류하는 새로운 인사이트를 발견할 수 있어, 수집한 빅데이터에 대한 정보가 부족할 때, 여건과 비용이 허락한다면 반드시 적용하는 방법론의 하나이다.

- 정의
  - 데이터 마이닝 기술의 한 방법으로 빅데이터에서 데이터의 특성을 고려하여 군집을 정의하고 대표점을 찾는 작업
- 빅데이터 시대의 의미
  - 데이터를 분류하는 데에 도움
  - 새로운 정보를 발견하는 실마리

● 군집의 구분

군집은 크게 계층적 군집화와 분할적 군집화로 구분할 수 있다. 계층적 군집화에서는 데이터의 점을 하나의 군집으로 설정하고 점 간의 거리를 기반으로 분할/합병한다. 예를 들어 계통도 등은 그러한 한 방법의 갈래라 볼 수 있다. 한편, 분할적 군집화는 여러 개의 분할 기법을 결정하는 기법이다. 다양한 거리 및 평가 함수에 기반하여 작동한다. 대표적으로는 -Means 알고리즘 등이 알려져 있다.

- 계층적 군집화
  - 데이터의 점을 하나의 군집으로 설정하고 점 간의 거리를 기반으로 분할/합병
  - 예) 계통도 등을 통하여 유사성 확인 가능
- 분할적 군집화
  - 여러 개의 분할 기법을 결정하는 방법
  - 거리 함수 및 평가 함수에 기반

- 예) k-Means 알고리즘 등

● 계통도Dendrogram

계통도를 이용하면 각 계층에서 군집의 유사성을 쉽게 확인할 수 있다. 계통도는 흡수 과정과 분리 과정으로 구분된다. 흡수 과정에서는 아래에서 위 방향으로 처리하여 군집을 흡수한다. 반면, 분리 과정은 위에서 아래로 분리하는 과정이다.

- 개요
  - 각 계층에서 군집의 유사성을 쉽게 확인할 수 있다.
- 흡수 과정 Agglomerative
  - 아래에서 위로 처리하여 군집을 흡수
  - n개의 각 군집과 수열의 형태가 연속적인 흡수 군집화 과정으로 처리
- 분리 과정Divisive
  - 위에서 아래로 분리하는 과정
  - 하나의 군집에 개의 표본이 있으며, 연속적인 분리 과정으로 수행

● k-Means 알고리즘

k-Means 알고리즘은 미지의 빅데이터로부터 k개의 분할 영역, 즉, 군집을 결정하는 방법이다. 이러한 군집을 결정하기 위해서는 적절한 거리 함수를 선정하여, 이에 기반하여 분할 영역을 탐색하는 것이 중요하다.

- 개요
  - k개의 분할 영역(군집)을 결정하는 방법
  - 거리 함수에 기반하여 분할 영역 탐색
- 수행과정
  - 1. 군집의 개수 k를 설정하고 군집의 초기값으로 중심을 1개씩 할당
  - 2. 주어진 중심점을 기준으로 하여 각 데이터를 가장 가까운 군집에 할당
  - 3. 할당된 데이터를 중심으로 각 군집은 새로운 중심점을 계산
  - 4. 새로운 중심점이 기존의 중심점과 차이가 없으면 이 단계에서 종료하고, 차이가 있는 경우 2번 단계로 되돌아가서 계속하여 수행

11강. 비정형 데이터 마이닝

1. 텍스트 마이닝

비정형 데이터의 한 갈래로 텍스트 데이터가 있다. 텍스트 데이터로부터 유의미한 정보를 추출하여 가치를 창출하기 위하여 텍스트 마이닝 방법론을 적용할 수 있다. 본 장에서는 텍스트 마이닝에 대하여 정리하여 보고, 텍스트 마이닝의 절차와 핵심 방법론에 대하여 학습하여 보도록 하자.

● 텍스트 데이터

텍스트 데이터는 ASCII나 UTF-8 등의 인코딩으로 표현된 데이터이다. 텍스트 데이터는 대부분 비정형 혹은 반정형인 경우가 많다. 이에, 자연어 처리(NLP)에 기반하여 정보를 추출하는 것이 필수이다.

- 개요
  - Text Data
  - ASCII, UTF-8 등의 인코딩 표현
- 특징
  - 비정형 혹은 반정형의 데이터인 경우가 많음
  - 자연어 처리(NLP: Nautral Language Processing)에 기반하여 정보를 추출

● 텍스트 마이닝 절차

텍스트 마이닝은 데이터 수집, 데이터 처리, 데이터 추출, 데이터 분석 등의 과정을 거친다. 각 과정에 대한 상세는 다음과 같다:

- 데이터 수집
  - 비정형/반정형 텍스트 데이터를 수집
- 데이터 처리
  - 특정 키워드나 의미있는 요소를 추출
  - 전처리(preprocessing) 수행
- 데이터 추출
  - 수학적 모델이나 알고리즘으로 정보 추출
  - NLP,TF-IDF 등의 방법 사용

- 데이터 분석
  - 최종 키워드, 의미있는 요소의 우선순위를 도출하는 단계

● 자연어 처리Nautral Language Processing

자연어 처리(NLP)는 인간의 언어 현상을 컴퓨터에서 모사할 수 있도록 연구하고 구현하는 인공지능의 주요 분야로, 텍스트 마이닝의 핵심 도구이기도 하다. 자연어 처리를 통하여 형태소 분석, 품사 부착, 구절 단위 분석, 구문 분석, 어휘 분석 등이 가능하다.

- 정의
  - NLP라고도 부르며, 인간의 언어 현상을 컴퓨터에서 모사할 수 있도록 연구하고 구현하는 인공지능의 주요 분야
- 작업
  - 형태소 분석 / 예) 나는 = 나(대명사)+ 는(조사)
  - 품사 부착 – 적절한 품사를 부착하여 문장 완성
  - 구절 단위 분석 – 명사구/동사구/부사구 등
  - 구문 분석, 어휘 분석

● 어휘 분석 (Lexical Analysis)의 과정

자연어 처리의 어휘 분석 과정은 문장 분리, 토큰화, 형태소 분석, 포스 태깅 등의 과정을 거친다. 각 과정의 상세를 정리하면 다음과 같다:

- 문장 분리Sentence Splitting
  - 말뭉치(corpus)를 문장 단위로 분리
  - 마침표(.) 등의 기호를 이용하여 분리
- 토큰화Tokenize
  - 토큰(token)은 의미를 가진 문자열
  - 이후 작업을 위하여 토큰으로 분리
- 형태소 분석Morphological Analysis
  - 단어의 수를 줄여 분석의 효율성을 높임
  - 예) cars와 car, stopped와 stop, POS(Part-Of-Speech)
- 포스 태깅POS Tagging
  - 토큰의 품사 정보를 할당하는 작업

- TF-IDF

TF-IDF는 여러 문서로 이루어진 문서군에서 출현하는 특정 단어가 문서 내에서 얼마나 중요한지 그 중요한 정도를 표현하기 위한 통계적 수치이다. 그 이름에서부터 알 수 있듯이, TF-IDF는 TF치와 IDF치를 구하고 이들을 곱함으로써 계산할 수 있다.

- 정의
  - 여러 문서로 이루어진 문서군에서 출현하는 특정 단어가 문서 내에서 중요한 정도를 표현하는 통계적 수치
  - TF와 IDF의 곱으로 표현

- TF (Term Frequency)

TF는 문서 내에서 특정 단어가 출현하는 빈도를 나타낸다. 말 그대로 문서의 단어 측정 빈도를 측정하여 기록하게 된다. TF치를 통하여 특정 단어의 출현 빈도가 높을수록 해당 단어의 중요도가 다른 단어에 비하여 상대적으로 증가하게 된다.

- 정의
  - 문서 내에서 특정 단어의 출현 빈도
- 방법과 특징
  - 문서 내의 단어의 측정 빈도를 측정
  - 자주 등장할수록 → 해당 단어의 중요성 증가

- IDF (Inverse Document Frequency)

IDF는 문서군 내에서 등장하는 단어의 빈도를 나타낸다. 문서가 아닌 문서군이므로, 여러 문서에서 특정 단어가 얼마나 등장하는지 지표를 계산하여야 한다.

- 정의
  - 문서군 내에서 등장하는 단어의 빈도

2. 멀티미디어 마이닝

비정형 데이터 마이닝의 한 갈래로, 멀티미디어 빅데이터에 대한 데이터 마이닝 기법의 중

요성이 증대하고 있다. 특히, 텍스트 마이닝과 다르게, 멀티미디어 마이닝은 그 분석 기법이 정형화되어 있지 않으며, 데이터의 종류 또한 무수히 많다. 향후 빅데이터 분석에 있어 멀티미디어 마이닝의 방법론을 설계하는 것이 주요 과업이 될 수도 있다. 본 장에서는 이처럼 중요한 멀티미디어 마이닝의 기법에 대하여 학습하여 보도록 한다.

● 멀티미디어(Multimedia)의 정의

멀티미디어는 다양한 매체로 정의할 수 있다. 오디오, 이미지, 비디오, 뉴미디어 등으로 구분할 수 있다. 각 미디어의 상세는 다음과 같다:

- 오디오(Audio)
  - 소리를 디지털 데이터 형태로 저장
  - 예) WAV, MP3, AAC 등의 파일 형식
- 이미지(Image)
  - 시각 정보를 디지털 데이터 형태 저장
  - 예) JPEG, PNG 등의 파일 형식
- 비디오(Video)
  - 소리와 시각 정보를 시간의 흐름에 따라 저장한 데이터 형식 / 예) MPEG4,HEVC
- 뉴미디어(New Media)
  - 위의 유형으로 정의할 수 없는 멀티미디어 데이터 형식

● 멀티미디어 마이닝 방법론

멀티미디어 마이닝은 다른 데이터 마이닝과 마찬가지의 프로세서를 거칠 수 있으나, 일반적인 데이터와 다르게 전처리 단계를 상세히 정의하여야 한다는 특징을 가지고 있다. 특징 추출에서는 사람의 힘으로 데이터를 분석하고 이에 따라 특징을 추출한다. 반면 딥 러닝 기반 방법에서는 전처리 된 멀티미디어 데이터에 대하여 인공신경망 등의 방법론을 적용하여 자율학습하고 특징을 추출한다.

- 특징 추출(Feature Extraction)
  - 멀티미디어 데이터를 사람의 힘으로 분석하고 이에 따라 특징(feature)을 추출하는 방법
- 딥 러닝(Deep Learning)
  - 전처리 된 멀티미디어 데이터에 대하여 인공신경망(ANN) 등의 방법론을 적용하

여 자율학습하여 특징을 추출하는 방법

● **멀티미디어 특징 추출의 예시**

오디오, 이미지, 비디오, 뉴미디어와 같은 다양한 형태의 멀티미디어 데이터로부터 다양한 특징을 추출할 수 있다. 여기서는 널리 사용되는 일부의 예시를 알아보도록 하자.

- 오디오(Audio)
  - 푸리에(Fourier) 변환을 통한 정보 추출
- 이미지(Image)
  - JPEG 등의 기반 기술인 DCT, DWT 등을 적용하여 특징 추출
- 비디오(Video)
  - 인터(inter) 및 인트라(intra) 특징 추출
  - H.264, HEVC 등의 코덱 특징에 기반
- 뉴미디어(New Media)
  - 오디오, 이미지, 비디오의 특성 활용
  - 현장, 무대, 설치 등의 특성 활용

3. 소셜 네트워크 마이닝

스마트 시대에 진입하며 빅데이터 데이터는 소셜 네트워킹 서비스로부터도 생성되고 있다. 이러한 데이터로부터 파악할 수 있는 정보의 양은 무궁무진하다. 본 장에서는 소셜 네트워크를 통하여 비정형 데이터를 마이닝하기 위한 일반적인 방법론에 대하여 학습하여 본다.

● **소셜 네트워킹 서비스(SNS)**

소셜 네트워킹 서비스는 사용자 간의 자유로운 의사소통과 정보 공유, 인맥 확대를 가능케 하며, 이를 통하여 사회적 관계를 생성하고 강화하여 주는 온라인 플랫폼이다. 소셜 네트워킹 서비스를 통하여 사회적 관계망을 생성하고 유지하며, 강화하고 확장하는 것이 가능하다. 특히 최근의 SNS 서비스는 대부분 웹 기반인 점을 감안할 때 다양한 유의미하며 가치 있는 정보를 내포하고 있음을 짐작할 수 있다. 향후 빅데이터 분석을 통한 마케팅의 활용 가치가 무궁무진할 것이다.

- 정의

- 사용자 간의 자유로운 의사소통과 정보 공유, 인맥 확대 등을 통하여 사회적 관계를 생성하고 강화하여 주는 온라인 플랫폼

- 특징
  - 사회적 관계망을 생성, 유지, 강화, 확장 가능
  - 최근의 SNS 서비스는 대부분 웹 기반
  - 빅데이터 분석을 통한 마케팅 활용가치 높음

● 소셜 마이닝(Social Mining)

소셜 네트워킹 서비스를 통한 데이터 마이닝을 가리켜 소셜 마이닝이라고 한다. 소셜 마이닝에서 수행하는 작업은 다음과 같다.

- 문서 수집Crawling
  - SNS 등을 통하여 사용자의 문서 수집
  - 사용자의 저작권 및 개인정보 유의
- 필터링Filtering
  - 스팸 데이터, 무관 문서 등을 필터링
  - 연관문서를 토대로 분석
- 자연어처리 분석NLP Analysis
  - 자연어처리 방법론의 다양한 기법을 활용하여 연관어 분석
- 데이터 분석 보고Reporting
  - 분석 결과를 시각화하고 해석하여 보고

12강. 데이터 시각화

1. 시각화의 정의와 개념

● 시각화 (Visualization)

데이터 시각화는 특정한 기준에 따라 분석한 데이터의 특징이나 분석 결과를 쉽게 이해할 수 있도록 그림/그래프 등으로 표현하여 주는 기술이다. 따라서 데이터 시각화를 다른 말로 데이터 표현이라고도 한다.

빅데이터 시대에는 데이터 시각화를 통하여 다양한 정보가 파급효과를 일으킬 수 있는 촉매가 될 수 있으므로, 가장 중요한 마지막 단계라고 볼 수 있다.

- 정의
  - 특정한 기준에 따라 분석한 데이터의 특징이나 분석의 결과를 분석가 및 사용자들이 쉽게 이해할 수 있도록 그림이나 그래프 등으로 표현하여 주는 기술
  - 표현(Representation)이라고도 함
- 빅데이터 시대의 의미
  - 유의미한 정보가 파급효과를 일으킬 수 있도록 하는 가장 중요한 마지막 단계

● 시각화 절차

시각화 절차는 정보 조직화 단계, 정보 시각화 단계, 상호작용 단계로 이어진다. 각 단계에 대한 작업과 설명은 다음과 같다:

- 정보 조직화 단계
  - 사용자의 정보 인지에 관여하는 단계
  - 혼돈의 상태로 존재하고 있는 데이터를 분류, 배열, 조직화하여 질서를 부여하는 단계
- 정보 시각화 단계
  - 사용자의 정보 지각에 관여하는 단계
  - 보다 효율적으로 정보 전달을 하기 위하여 인간의 오감(시각/청각/촉각/미각/후각)과 관련된 감각 기관에 최적의 자극을 부여하는 방법 제시
- 상호작용 단계
  - 정보와 사용자 간의 상호작용 측면으로서 사용자 경험(UX)을 디자인하는 단계
  - 정보의 인지적 요인 뿐만 아니라 지각적 요인을 함께 활용하는 단계

- 정보 시각화 단계와 밀접하게 연동되면서 동시에 입력 기술의 특성 또한 함께 고려

2. 시각화 방법론

● 시각화 기술

다양한 시각화 기술을 이용하여 빅데이터 분석의 결과를 표현할 수 있다. 본 장에서는 Tag Cloud, GraphViz, Processing, Tableau, Gephi 등 시각화 기술에 대하여 소개한다.

- Tag Cloud
  - 태그 연관성에 따른 빈발도 및 관계 분석
- GraphViz
  - 흐름도나 트리 다이어그램 생성 표현 도구
- Processing
  - 그래픽 디자인을 위한 프로그래밍 언어
- Tableau
  - 데이터의 시각적 분석과 보고 도구 제공
- Gephi
  - 데이터를 네트워크 형태로 생성 후 표현

● Tag Cloud
- 메타데이터에서 얻은 태그를 다양한 기준으로 분석하여 표현
- 대부분 웹페이지 또는 이미지 표현, 2차원 표 형태로 배치
- 각 태그들의 인기도를 한 눈에 파악 가능

● Gephi
- 가공되지 않은 그래프 데이터를 네트워크로 생성
- 사용자가 데이터 및 노드의 위치를 자유롭게 수정 및 조절 가능
- 기본적인 노드 표현 예시 제공

● GraphViz
- 데이터를 기반으로 다이어그램을 그릴 수 있는 명령어 라인 네트워크 그래프 시각화 도구로, 흐름도와 트리 표현에 활용 가능
- 다양한 디자인 옵션을 제공하여 커스터마이즈 가능

● Processing

- 디자이너와 아티스트의 프로그래밍 접근도를 높여주는 IDE를 이용하여 손쉽게 그래픽 표현을 구현할 수 있는 프로그래밍 환경
- 다양한 함수를 통하여 커스터마이즈 된 그래픽 표현 가능

- Tableau
  - 그래프를 시각화하는 데스크톱 응용 프로그램
  - 전문적인 출판물에 첨부할 다양한 형태의 그래프 생성에 용이

3. 시각화 적용 사례

- **정치**
  - 국회의원의 본회 표결 데이터를 이용하여 국회 속기록을 바탕으로 한 데이터 시각화
- **경제**
  - OECD 회원국의 데이터를 비교하여 회원국의 삶의 질을 지표화
- **공공서비스**
  - 시민에게 공공데이터를 오픈하고 시각화에 활용

13강. 데이터 시각화 도구

1. R의 활용

앞서 배운 데이터 시각화의 절차와 사례를 통하여 다양한 데이터 시각화 도구를 고려할 수 있다. 본 장에서는 가장 범용적이고 초기적인 단계로 적용할 수 있는 데이터 시각화 도구로 R의 데이터 시각화 도구로의 응용을 학습해보고자 한다.

● R을 데이터 시각화 도구로 활용하기

R은 데이터 처리, 분석 뿐만 아니라 통계 분석, 그래픽 표현, 보고 작성을 위한 프로그래밍 언어이자 소프트웨어 환경으로 볼 수 있다. 특히 그래픽 표현 및 보고 작성 기능을 가지고 있으며, 이를 이용하여 막대형 그래프, 원형 그래프, 3차원 그래프, 히스토그램 등 다양한 형태의 그래프를 출력할 수 있다.

- 정의
  - 통계 분석, 그래픽 표현, 보고 작성을 위한 프로그래밍 언어 및 소프트웨어 환경
  - 그래픽 표현 및 보고 작성
  - 막대형 그래프, 원형 그래프, 3차원 그래프,히스토그램 등 다양한 형태의 그래프 출력

이후에는 2차원 그래프의 표현 방법에 대하여 학습하여 보도록 한다.

● 2차원 그래프의 표현

R에서는 plot() 함수를 사용하여 2차원 그래프를 표현할 수 있다. plot() 함수를 이용하면 x축 데이터와 y축 데이터를 지정하고, 다양한 옵션을 이용하여 제목, 범례, x/y축 레이블 등을 표현할 수 있다.

R에서 plot() 함수의 문법과 옵션은 다음과 같다.

- 문법

```
plot(<y축 데이터>,<옵션1>,<옵션2>,…)
plot(<x축 데이터>,<y축 데이터>,<옵션1>,<옵션2>,…)
```

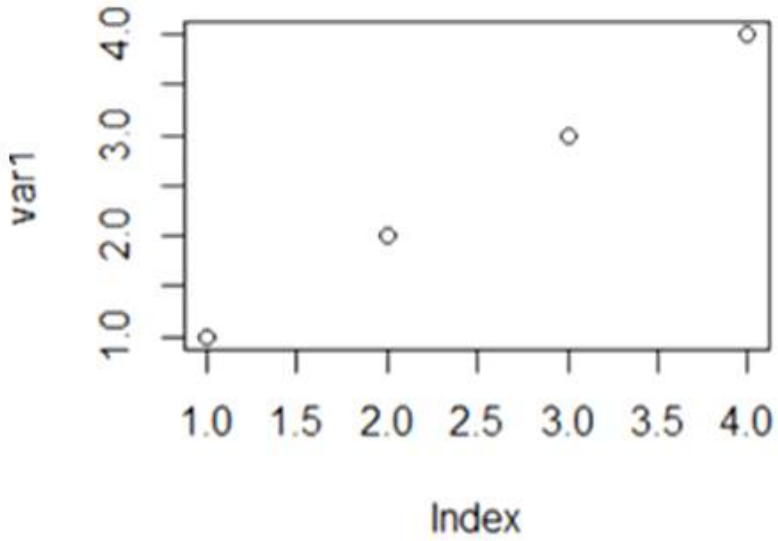
- 옵션

옵션	설명
main="<value>"	그래프 제목 설정
sub="<value>"	그래프 보조 제목 설정
xlab="<value>" ylab="<value>"	x, y 축의 레이블 설정
type="<value>"	그래프의 모양 설정 p : 점 모양 그래프 (기본값) l : 선 모양 그래프 (꺾은선)
lty="<value>"	blank : 투명선 solid : 실선 dashed : 대쉬선 dotted : 점선
col="<value>"	색상

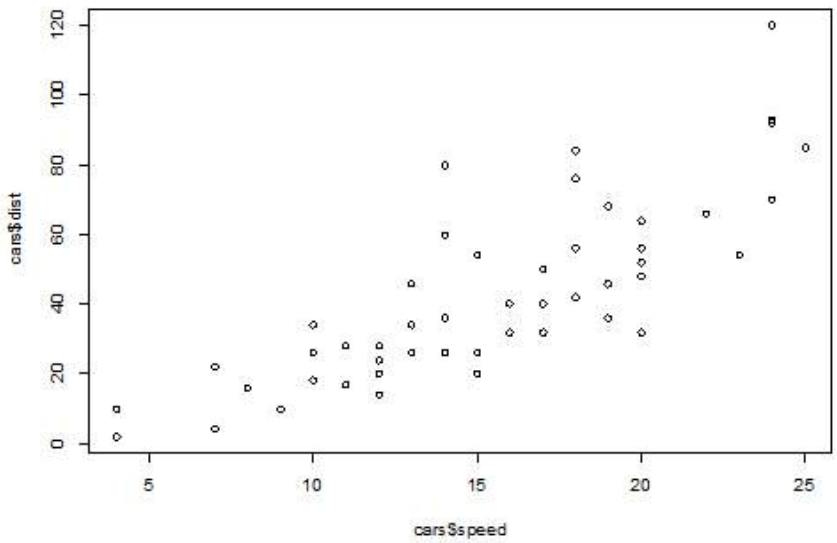
R에서 2차원 그래프를 표현하기 위한 몇 가지 예시를 제시하면 다음과 같다. 먼저 첫 번째 예시는 (1,1), (2,2), (3,3), (4,4)를 2차원 그래프로 표현하기 위한 예시이다. 이때, 선을 별도로 지정해주지 않을 경우, 기본 마커(o)로 그래프가 표현된다. 두 번째 예시는 cars 데이터셋으로부터 산점도를 표현하는 그래프이다. cars$speed는 x축, cars$dist는 y축을 표현한다. 이를 이용하여 데이터의 분포가 전반적으로 어떻게 되는지 조망할 수 있다.

- 예시

```
> var1 <- c(1,2,3,4)
> plot(var1)
```




```
> names(cars)
## [1] "speed" "dist"
> plot(cars$speed, cars$dist)
```



- **히스토그램의 표현**

R에서는 데이터의 범주를 계급으로 나누고 이를 기반으로 데이터의 빈도를 표현하는 히스

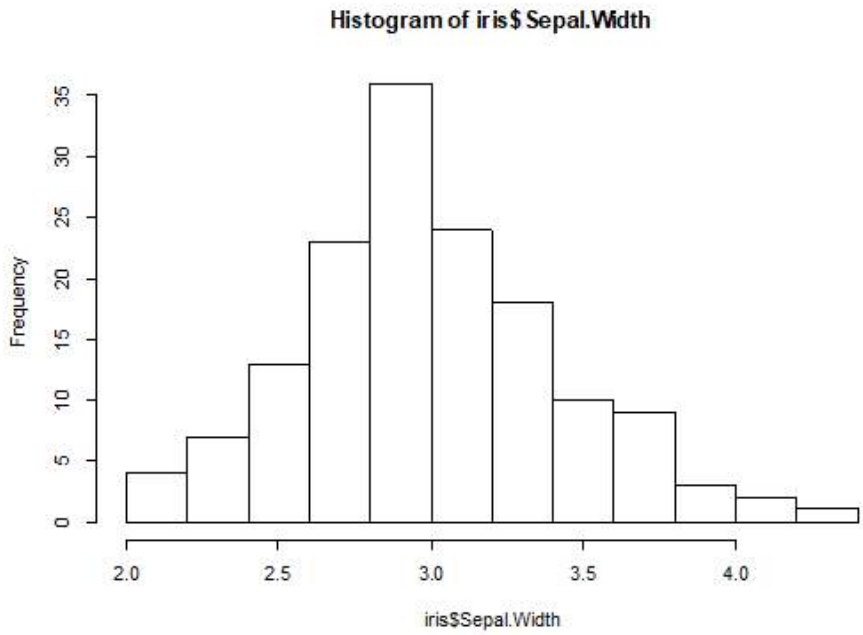
토그램을 그릴 수 있다. 본 장에서는 예시를 통하여 히스토그램을 그리는 방법에 대하여 학습한다.

```
hist(<벡터 데이터>, <옵션1>, <옵션2>, …)
```

예시에서는 iris 데이터셋의 iris$Sepal.Width를 히스토그램으로 표현하고 있다. hist( ) 함수에 아무런 옵션을 지정하지 않을 경우 기본적으로 12개의 계급으로 히스토그램이 표현된다. hist( ) 함수의 다양한 옵션을 이용하여 히스토그램의 형태를 다양하게 변경할 수 있다.

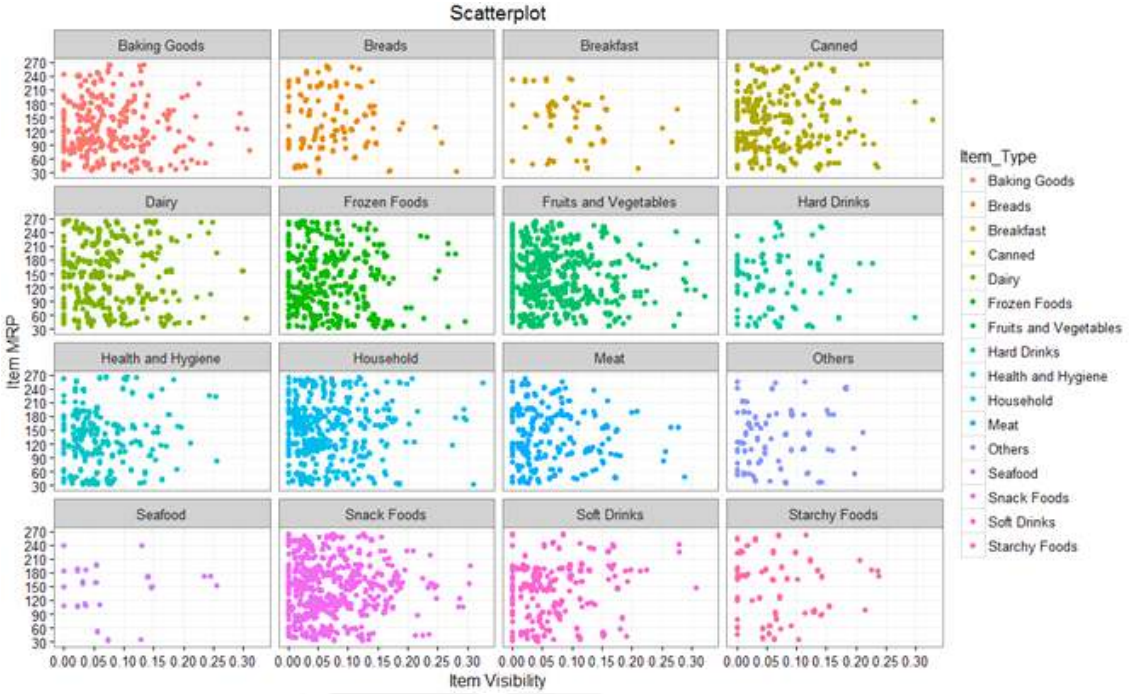
- 예시

```
> hist(iris$Sepal.Width)
```

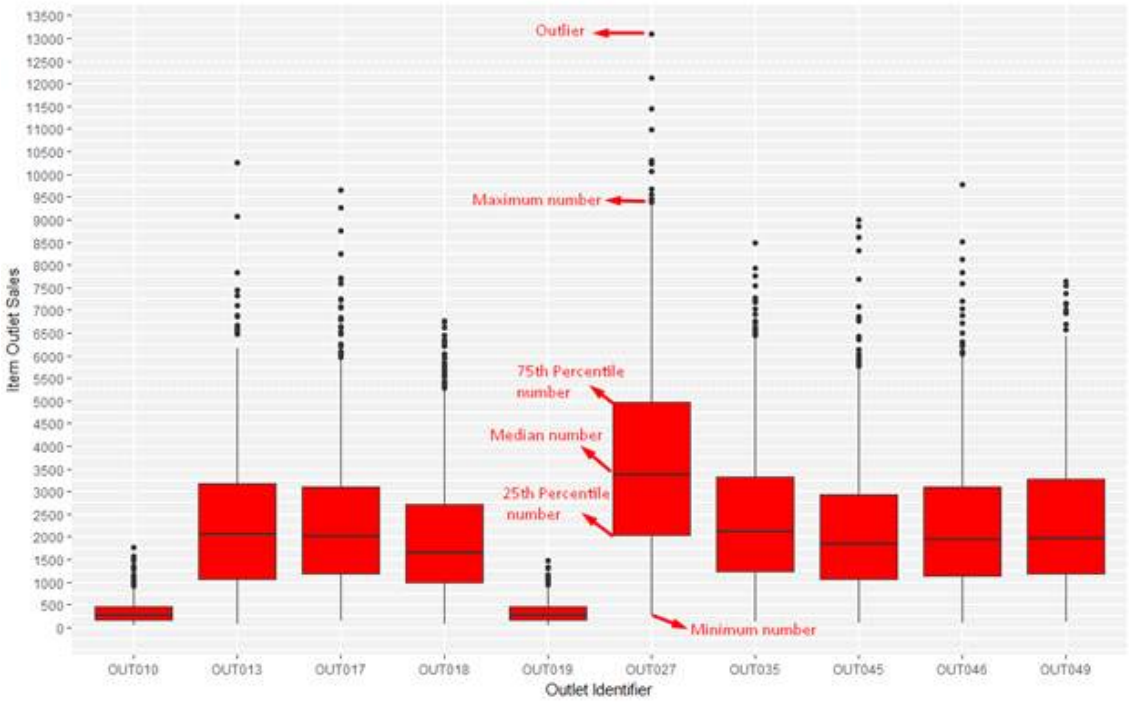


- **R을 사용한 데이터 시각화 사례**

- ggplot의 산점도(Scatterplot)를 사용한 항목별 표현



- ggplot의 boxplot을 사용한 데이터 시각화 표현 사례



2. D3.js

- **D3.js의 개요**

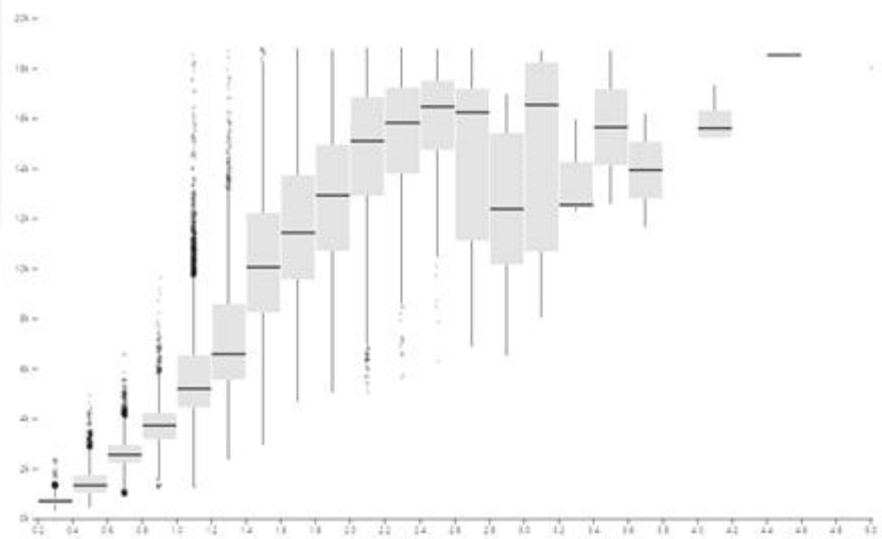
D3.js는 웹브라우저 상에서 작동하는 동적이고 인터랙티브한 정보 시각화 구현을 위한 자바

스크립트 라이브러리이다. D3.js를 이용하여 선택, 변화, 배열, 수식, 색, 스케일, SVG, 시간, 레이아웃, 지오그래피/지오메트리, 행위 등 다양한 형태로 시각화하는 것이 가능하다.

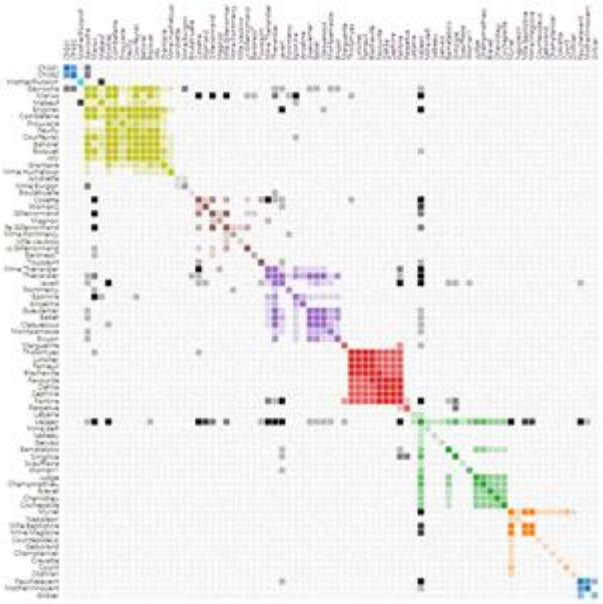
- 정의
  - 웹브라우저 상에서 동적이고 인터랙티브한 정보시각화를 구현하기 위한 자바스크립트 라이브러리
- 기능의 분류
  - 선택, 변화, 배열, 수식, 색, 스케일, SVG,시간, 레이아웃, 지오그래피/지오메트리, 행위
- 특징
  - D3.js는 별도의 프로그램이 아닌 자바스크립트로 작동
  - 따라서 자바스크립트를 통한 프로그래밍 필요
  - 결과물은 웹 브라우저를 통해서 볼 수 있음

● D3.js의 사례

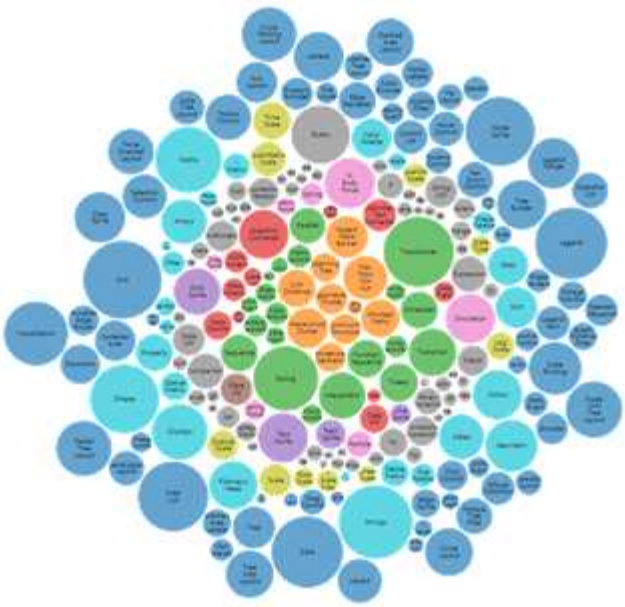
- 박스 플롯(Box Plot)



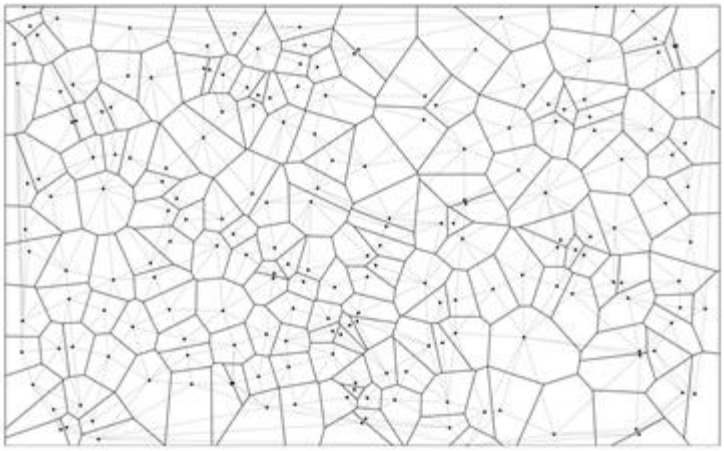
- 동시출현행렬(Co-Occurrence Matrix)



- 버블 차트(Bubble Charts)

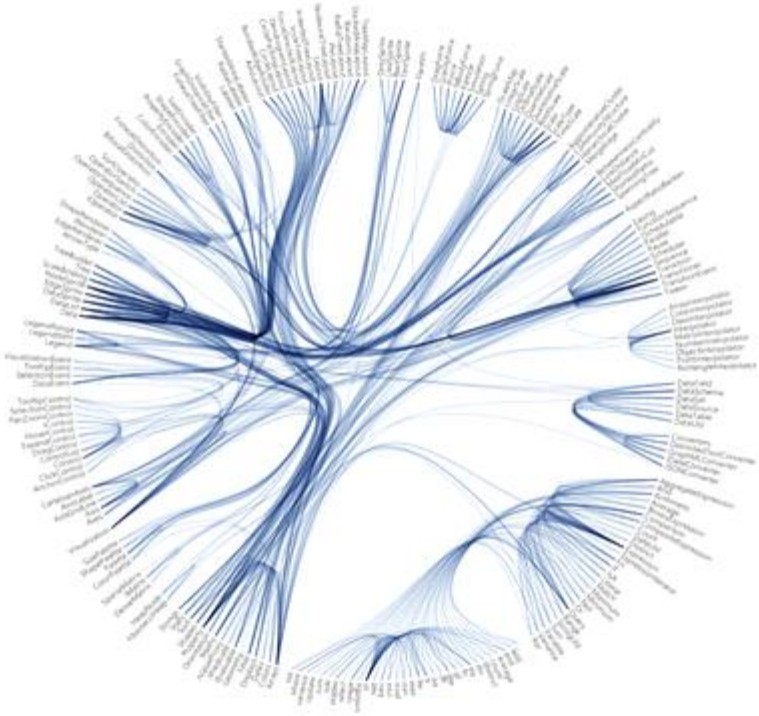


- 보로노이 다이어그램(Voronoi Diagrams)



- 계통도(Dendrogram)

- 계층적 엣지 번들링 (Hierarchical Edge Bundling)



14강. 빅데이터 프로젝트 기획과 관리

1. 빅데이터 프로젝트 기획과 관리

본 장에서는 빅데이터 프로젝트를 기획하고 관리하는 일체의 방법에 대하여 알아본다. 먼저 프로젝트 관리에 대하여 정의하고, 빅데이터 프로젝트에 대하여 정의하여 보도록 한다.

● 프로젝트 관리(Project Management)

프로젝트 관리란, 프로젝트의 성공적인 완성을 목표로 행하는 일체의 활동을 뜻한다. 프로젝트 관리에는 프로젝트 활동계획, 일정표, 진척 관리 등을 포함한다. 프로젝트 관리에 필요한 적절한 방법론을 적용하여 기술 전달 및 표준화를 꾀하고 효율적인 업무를 수행하는 것이 가능하다.

- 정의
  - 프로젝트의 성공적인 완성을 목표로 행하는 일체의 활동
  - 프로젝트 활동계획, 일정표, 진척 관리 포함
- 의미
  - 관리 방법론을 적용하여 기술 전달 및 표준화를 꾀하고 효율적인 업무 수행 가능

● 프로젝트 관리 활동

프로젝트 관리 활동에는 프로젝트 기획, 리스크 측정, 자원 산정, WBS 작성, 자원 확보, 비용 산정, 작업 할당, 진척 관리, 결과 분석 등 다양한 작업을 포함한다.

- 프로젝트 기획
- 위험(risk) 측정
- 이용 가능 자원 산정
- 작업 분류 체계(WBS) 작성
- 인적/물적 자원 확보
- 인적/물적 비용 산정
- 작업 할당
- 진척 관리
- 결과 분석

● 빅데이터 프로젝트

빅데이터 프로젝트 또한 프로젝트의 일종으로, 차이점이 있다면, 빅데이터에 대한 방법론을 적용하여 프로젝트의 완수를 위하여 행한다는 점이다. 빅데이터 프로젝트의 주요 특징은, 데이터 분석 결과에 따라 중도에도 방향 전환이 가능하며, 일회성 소스코드 산출물이 다량 발생하고, 빈번한 보고서 산출물 또한 발생한다는 점이다. 따라서 일반적인 프로젝트와 달리, 빅데이터 프로젝트는 그 관리를 위하여 조금 다른 접근 방식이 필요하다.

- 정의
  - 빅데이터에 대한 방법론을 적용하여 프로젝트의 완수를 위하여 행하는 일체의 활동
- 특징
  - 데이터 분석 결과에 따라 방향 전환 가능
  - 일회성 소스코드가 다량 발생
  - 빈번한 보고서 산출물 발생

2. 빅데이터 프로젝트 기획 방법론

● 빅데이터 프로젝트 기획

빅데이터 프로젝트 기획이란, 빅데이터 프로젝트의 목표 달성을 최적화하기 위하여 의사결정 및 정보와 인사이트를 과학적 분석으로 제공하는 분석 체계이다.

- 정의
  - 목표 달성을 최적화하기 위하여 의사결정 및 정보와 인사이트를 과학적 분석을 통해 제공하는 분석체계

● 빅데이터 프로젝트 기획 단계

빅데이터 프로젝트 기획은 크게 분석 단계와 계획 단계로 나뉜다.

- 분석 단계
  - 어떤 문제를 해결 목적과 이유에 대하여 정의
  - 문제나 기회를 탐색하고 과제를 도출
- 계획 단계

- 분석 단계에서 정의된 문제를 해결하는과정에 대하여 계획 및 설명

● 빅데이터 프로젝트 기획 중 분석 단계

빅데이터 프로젝트 기획 중 분석 단계에서는 문제 정의, 문제 발굴, 대안 설계, 타당성 검토를 통하여 과제를 선택하게 된다. 각 상세는 다음과 같다:

- 문제 정의
  - 사용자 관점에서 문제를 정의
  - 예) 고객 이탈의 요인 및 관련성, 이탈 예측 등
- 문제 발굴
  - 기회 식별 및 문제 식별을 수행
  - 예) 고객 이탈 현상 심화, 납기 지연 및 손실 초래
- 대안 설계
  - 문제 해결을 위한 다양한 정의 모색
  - 예) 기존 정보 시스템 활용, 빅데이터 분석 등
- 타당성 검토
  - 제시된 대안에 대한 타당성 평가
  - 예) 경제적/기술적/운영적 타당성
- 과제 선택
  - 가장 우월한 대안을 선택하여 프로젝트화

● 빅데이터 프로젝트 기획 중 계획 단계

빅데이터 프로젝트 기획 중 분석 단계 이후에는 계획 단계에 돌입한다. 계획 단계에서는 목표를 정의하고, 요구사항을 도출하며, 예산안과 관리 계획을 수립하는 과정을 거치게 된다.

- 목표 정의
  - 과제 추진의 성공적인 달성을 위하여 성과 목표를 명확하게 정의
- 요구사항 도출
  - 데이터 및 기술지원 요구사항을 사전에 명확하게 정의
- 예산안 수립
  - 전 과정에 소요되는 예산을 검토하고 확보 계획을 수립하는 단계
- 관리 계획 수립

- 프로젝트 범위, 예산, 품질, 일정 등 주요 가정과 의사결정을 문서화

3. 빅데이터 프로젝트 관리 방법론

● 빅데이터 프로젝트 관리자

빅데이터 프로젝트 관리자란, 빅데이터 프로젝트를 성공적으로 완수하기 위하여 프로젝트의 계획, 수립, 수행을 책임지고, 관리 및 기술상의 문제를 해결하는 실무자이다.

- 정의
  - 빅데이터 프로젝트를 완수하기 위하여 프로젝트의 계획/수립/수행을 책임지며, 관리 및 기술상의 문제를 해결하는 실무자

● 빅데이터 프로젝트 관리자의 자질

빅데이터 프로젝트 관리자는 관리 능력, 기업가 능력, 지휘 능력, 의사소통 능력, 기술적 능력 등의 자질을 필수로 한다. 각 자질의 상세는 다음과 같다.

- 관리 능력
  - 예산, 인적/물적 자원, 일정 등에 대한 지식과 경험 및 관리 능력 필요
- 기업가 능력
  - 계획에 따라 최적의 비용으로 프로젝트 수행
- 지휘 능력
  - 프로젝트 팀원의 협력을 이끌 리더십 필요
  - 팀원의 전문성과 열정을 최대한 이끌어 낼 지휘 능력과 리더십이 필요
- 의사소통 능력
  - 프로젝트 팀원 및 관계 부서 간조정 및 의사소통을 위한 능력 필요
- 기술적 능력
  - 프로젝트 수행을 보조하고 문제 해결을 위한 기술적인 능력 필요

● 빅데이터 프로젝트 관리 수행

빅데이터 프로젝트의 관리 수행에는 다음과 같은 절차가 포함된다.

- 프로젝트 관리 추진체계 구성 및 역할 부여
  - 프로젝트 계획을 수행하고 목표 달성을 위한 적절한 조직을 구성하고 역할을 부여
  - 제조업/사무업 기반의 통상적 조직과 달리 수평적인 조직 문화가 필요
- 프로젝트 인적 자원 관리
  - 프로젝트 수행에 적절한 역량을 갖춘 인원
  - 각 인원의 역할과 책임을 명확히 규정
  - 인원 간의 의사소통 구조를 확립하고 관리
- 프로젝트 추진 일정 관리
  - 일정 단계 및 활동에 따라 소요 예상기간을 산정하여 일정 계획을 수립
  - 공동 업무 추진 시 공동 일정 계획 수립
  - Gantt 차트 등을 활용하여 적극적인 관리
- 프로젝트 실행 관리
  - 프로젝트의 정확한 실행 상황을 파악
  - 프로젝트 핵심 관리 구분마다 전체 계획에 대한 조정을 수행하는 한편, 계획 통합 및 문서화 작업 필요
  - 계획서에는 인적/물적 자원의 계획값, 진행 상황 파악 방법, 대응 및 통제 방안, 성과물 검수 방법, 완료 확인 방법 등을 포함

● 빅데이터 프로젝트 위험성 관리

한편, 빅데이터 프로젝트에도 위험성이 존재한다. 빅데이터 프로젝트의 위험성은 구현의 어려움, 구현 비용 및 구현 기간 관련 문제, 분석 시스템의 성능 등의 문제가 있다.

- 빅데이터 프로젝트에서의 위험성
  - 구현상 어려움 → 효과의 전부 혹은 일부를 상실
  - 구현비용이 계획보다 높은 경우
  - 구현기간이 계획보다 길어지는 경우
  - 분석 시스템의 성능이 기대에 미치지 못하는 경우

15강. 빅데이터 적용 사례

1. 정치, 경제, 사회 분야 적용 사례

- **정치 분야 적용 사례**
  - 2008년 미국 대통령 선거 사례
  - 민주당 측은 다양한 형태의 유권자 빅데이터 확보
  - 인적정보 : 인종, 종교, 나이, 가구형태, 소비수준 등
  - 성향정보 : 과거 투표 여부, 구독 잡지, 음료 취향 등
  - 개별 방문 및 소셜 미디어를 통한 정보 수집
  - 유권자 맞춤형 선거 전략 수립 및 시행 → 승리

- **경제 분야 적용 사례**
  - 아마존닷컴(amazon.com)의 마케팅 사례
  - 사용자의 구매내역을 데이터베이스에 기록 후 분석하여 소비자의 소비취향과 관심사를 파악
  - 고객별 추천 (Personalized Recommender System)
  - 롱테일(The Long Tail)현상에 주목

- **사회 분야 적용 사례**
  - 마이너리티 리포트(Minority Report)의 현실화
  - 생체 빅데이터 이용 → 시민 신원 및 동태 파악
  - 지역별 사회/경제 및 개인정보 빅데이터를 이용하여 범죄 발생 예측 시스템 구축
  - 개인정보 및 인권 침해의 한계점

- **2. 문화/스포츠 분야 적용 사례**
  - 2014년 FIFA 월드컵 독일 우승과 빅데이터
  - 독일 국가대표팀과 SAP 협력 → 'SAP 매치 인사이트'
  - 선수들에게 부착된 센서를 통하여 운동량, 속도, 심박수, 슈팅동작 등 비정형 데이터 수집
  - 빅데이터를 기반한 스포츠 운영 전술 수립

2. 과학, IT, 산업 분야 적용 사례

- **과학 분야 적용 사례**
  - 대학병원 의료장비 생성 스트림 데이터 → 10,000건/초
  - 빅데이터 기술로 실시간 통합 및 분석
  - 응급상황 조기예측 가능
  - 의사 및 간호사의 대기 및 노동 감소

- **IT 분야 적용 사례**
  - 자동차 센서 빅데이터로 스마트카 구축 (초당 1GB)

- 무인자동차 구현 → 기존 교통 시스템에 대혁명
- 자동차 보유자의 운용 비용 감소
- 운전에 소요되는 인력과 시간 소비 감소와 효율 증대

● **산업 분야 적용 사례**

- 반도체 공정 빅데이터(10TB/연 이상) 수집 및 분석으로 수율에 영향을 미치는 인자 및 설비 발견
- Tracing 분석 → 저수율 생산경로 분석 → 탐색 및 개선

3. 빅데이터의 미래

빅데이터의 미래에는 데이터 과학자(data scientist)가 등장할 것이며, 분석지능(AQ)이 중요해지는 시대가 될 것이다. 이들의 특성에 대하여 잘 이해하고 미래에 대비하는 것이 중요할 것이다.

● **데이터 과학자(Data Scientist)**

빅데이터 시대의 데이터 과학자는 데이터 분석으로 신가치를 창출하는 전문가이다. 데이터 과학자는 다양한 소스로부터 데이터를 수집하고, 복잡한 데이터를 구조화 및 단순화하며, 이상 데이터를 검색하고, 효과적인 모델링으로 분석 및 예측을 수행할 수 있다.

- 정의
  - 데이터 분석으로 신가치를 창출하는 전문가
- 하는 일
  - 다양한 소스로부터 데이터를 수집
  - 복잡한 데이터를 구조화 및 단순화
  - 이상 데이터 탐색
  - 효과적인 모델링으로 분석 및 예측 수행

● **분석지능 (AQ: Analysis Quotient)**

빅데이터 시대에는 분석지능(AQ)이 중요해질 것이다. 분석지능(AQ)은 빅데이터로부터 통찰력을 얻고 이를 바탕으로 미래를 예측하는 능력을 뜻한다. 분석지능(AQ)은 4단계로 표현할 수 있으며, 가급적 4번째 단계에 도달할 수 있도록 각 개인과 조직은 노력하는 것이 중요할 것이다.

- 정의
  - 빅데이터로부터 통찰력을 얻고 이를 바탕으로 미래를 예측하는 능력
- AQ 4단계

- 1단계 – 빅데이터 분석에 관심을 가짐
- 2단계 – 빅데이터를 단순 활용
- 3단계 – 빅데이터로 비즈니스 연계
- 4단계 – 빅데이터로 미래 예측 및 성과 창출

● 빅데이터의 미래

미래사회는 **불확실성, 리스크, 스마트, 융합** 등의 특성을 갖는 것으로 생각할 수 있다. 그러한 미래 시대에 빅데이터의 역할은 중요해진다고 할 수 있다. 미래사회의 4가지 특성 각각에 대해 빅데이터는 통찰력, 대응력, 경쟁력, 창조력으로 대응할 수 있다. 각 대응 방법의 특성은 다음과 같다:

- 미래사회의 특성
  - 불확실성, 리스크, 스마트. 융합
- 빅데이터의 역할
  - 통찰력, 대응력, 경쟁력, 창조력

- 통찰력
  - 사회현상, 현실 빅데이터 기반 패턴 분석
  - 다수의 가능한 시나리오로 통찰력 제시
- 대응력
  - 환경, 소셜, 모니터링 정보의 패턴 분석
  - 이슈를 사전 인지 및 분석, 빠른 의사결정
- 경쟁력

- 상황인지, 인공지능, 개인화/지능화 서비스
- 트렌드 변화 분석을 통한 경쟁력 확보

- 창조력

- 타분야와의 유연한 결합을 통한 가치창출
- 방대한 데이터를 통한 새로운 시장 창출

● 빅데이터의 성공조건

빅데이터가 성공하기 위해서는 데이터 기반 문화가 형성되고 적용되며, 분석지능(AQ)을 극대화하고, 분석결과와 혁신의 연결 및 지속이 이루어지는 것이 중요하다. 각 단계에 성공요건은 다음과 같다.

- 데이터 기반 문화 형성 및 적용

- 직관이나 비논리적 사고가 아닌 실제의 데이터에 기반한 의사결정 문화
- 과학적 의사결정 문화
- 리더의 혁신 의지가 중요

- 분석지능(AQ)을 극대화

- 분석지능은 혁신능력 및 리스크 관리 능력과 연관성이 높음
- 작은 업무에서부터 빅데이터 분석 시작
- 빅데이터 교육을 통한 빅데이터 의식 확산

- 분석결과와 혁신의 연결 및 지속

- 분석결과를 직접적인 혁신으로 연계하여 실질적인 이용이 이루어질 수 있도록 함
- 일회성의 사용이 아닌 지속적인 사용 도모
- 데이터 거버넌스 구축의 필요성