3과목 데이터 분석 기획

1장 데이터 분석 기획의 이해

제1절 분석 기획 방향성 도출

분석 기획 : 실제 분석을 수행하기에 앞서 분석을 수행할 과제의 정의 및 의도했던 결과를 도출할 수 있도록 이를 적절하게 관리할 수 있는 방안 사전에 계획하는 작업(What, Why, How)

1. 분석 기획의 특징

분석 기획 :해당 문제 영역에 대한 전문성 역량(Domain Knowledge)+수학/통계학적 지식 활용한 분석 역량(Math&Statistics)+분석 도구인 데이터 및 프로그래밍 기술 역량(information Technology)**+프로젝트 관리/리더십 역량**

분석 방법

- How&What :

How↓/ What→	Known	Unknown
Known	Optimization 최적화	Insight 통찰
Unknown	Solution 솔루션	Discovery 발견

- 목표 시점별→두가지 융합

• 과제 중심적인 접근 방식 : 당면 과제 단위로 빠르게 해결, Quick-Win 방식, Speed&Test

• 장기적인 마스터 플랜 방식 : 지속적인 분석 내재화(Accuracy&Deploy), 전사적인 관점(Long Term View), 분석과제 정의(Problem Definition)

2. 분석 기획 시 고려사항

1) 가용한 데이터(Available Data) : 데이터의 확보 필수, 데이터 유형에 따른 솔루션 및 분석 방법Ex) Transaction data, Human-generated data, Mobile data, Machine and sensor data

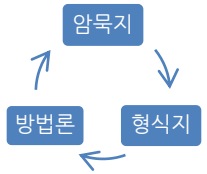
2) 적절한 유스케이스(Proper Business Use Case) : 분석을 통해 가치가 창출될 수 있는 적절한 활용 방안과 활용 가능한 유스케이스의 탐색, 기존의 유사 분석 시나리오 및 솔루션 있다면 최대한 활용Ex) Customer analytics, Social media analytics, Plant and facility management, Pipeline management, Price optimization, Fraud detection

3) 분석과제 수행시 장애요소(Low Barrier of Execution) : 장애요소에 대한 사전 계획 수립Ex) Cost, Simplicity, Performance, Culture, Change Management(교육 등 변화 관리)

제2절 분석 방법론

1. 분석 방법론 개요

고정관념(Stereotype), 편향된 생각(Bias), 프레이밍 효과(Framing Effect : 동일한 사건이나 표현 방식에 따라 개인의 판단이나 선택이 달라지는 경우) 등은 기업의 합리적 의사결정을 가로막는 장애 요소이다. 이를 개선하기 위해 데이터 기반의 의사결정 필요



방법론은 상세한 절차(Procedures), 방법(Methods), 도구와 기법(Tools&Techniques), 템플릿과 산출물(Templates&Outputs)로 구성되어 어느 정도의 지식만 있으면 활용 가능

- 폭포수 모델(Waterfall Medel) : 단계적으로 진행, 이전 단계가 완료되어야 다음 단계로 진행, 하향식(Top Down), 문제나 개선점 발견 시 전 단계로 돌아가는 피드백 과정 수행 가능

- 나선형 모델(Spiral Model) : 반복 통해 점증적으로 개발, 처음 시도하는 프로젝트 적용 용이

- 프로토타입 모델(Prototype Model) :본격적인 상품화에 앞서 성능을 검증·개선하기 위해 핵심 기능만 넣어 제작한 기본 모델

- 방법론의 구성 : 계층적 프로세스 모델(Stepwised Process Model) 아래 3계층으로 구성

• 단계(Phase) : 최상위 계층으로서 프로세스 그룹을 통해 완성된 단계별 산출물 생성, 각 단계는 여러개의 태스크로 구성되며 기준선(Baseline)으로 설정되어 관리되어야 하며 버전관리(Configuration Management) 등을 통해 통제**Process Group**

• 태스크(Task) :단계를 구성하는 단위 활동, 물리·논리적 단위로 품질 검토의 항목**Mapping**

• 스텝(Step) : WBS(Work Breakdown Structure)의 워크패키지에 해당되고 입력자료, 처리 및 도구, 출력자료로 구성된 단위 프로세스**Unit Process**

2. KDD 분석 방법론(Knowledge Discovery in Databases)

: 1996년 Fayyad가 체계적으로 정리한 데이터마이닝 프로세스, 데이터마이닝, 기계학습, 인공지능, 패턴인식, 데이터 시각화 등에서 응용될 수 있는 구조, 9개의 프로세스 제시

① 분석 대상 비즈니스 도메인의 이해
② 분석 대상 데이터셋 선택과 생성**(Selection)**
③ 데이터 포함된노이즈(Noise), 이상값(Outlier), 결측치(Missing Value) 제거하는 정제작업, 선처리=전처리**(Preprocessing)**
④ 목적에 맞는 변수를 찾고 필요시 데이터 차원을 축소하는 데이터 변경**(Transformation)**
⑤ 분석 목적에 맞는 데이터마이닝 기법 선택
⑥ 분석 목적에 맞는 데이터마이닝 알고리즘 선택
⑦ 데이터마이닝 시행**(Datamining)**
⑧ 데이터마이닝 결과에 대한 해석**(Interpretation/Evaluation)**
⑨ 데이터마이닝에서 발견된 지식 활용

3. CRISP-DM 분석 방법론(Cross Industry Standard Process for Data Mining)

: 1996년 유럽연합에서 시작, 99년 첫 버전 발표, 계층적 프로세스 4개 레벨, 6단계로 구성

Phase	
Generic Tasks	일반화 태스크,데이터마이닝의 단일 프로세스를 완전하게 수행하는 단위
Specialized Tasks	세분화 태스크, 일반화 태스크를 구체적으로 수행하는 단위
Process Instances	프로세스 실행, 데이터마이닝을 위한 구체적인 실행 포함

가. 업무 이해(Business Understanding): 상황 파악, 데이터마이닝 목표 설정

나. 데이터 이해(Data Understanding) :데이터수집, 품질 확인, 탐색 및기술 분석

다. 데이터 준비(Data Preparation) :데이터셋 선택, 데이터 정제, 통합, 포맷팅

라. 모델링(Modeling) : 모델 평가

마. 평가(Evaluation) : 분석 결과 평가, 모델링 과정 평가, 모델 적용성 평가

바. 전개(Deployment) : 프로젝트 리뷰, 모니터링·유지보수 계획 수립, 종료보고서 작성

4. 빅데이터 분석 방법론

분석 기획 (Planning)	데이터 준비 (Preparing)	데이터 분석 (Analyzing)	시스템 구현 (Developing)	평가 및 전개 (Deploying)
• 비즈니스 이해 및 범위 설정 • 프로젝트 정의 및 계획 수립 • 프로젝트 위험계획 수립	• 필요 데이터 정의 • 데이터 스토어 설계 • 데이터 수집 및 정합성 점검	• 분석용 데이터 준비 • 텍스트 분석 • 탐색적 분석 • 모델링 • 모델 평가 및 검증 • 모델 적용 및 운영 방안 수립	• 설계 및 구현 • 시스템 테스트 및 운영	• 모델 발전 계획 수립 • 프로젝트 평가 및 보고

5. 분석 계획(Planning)

분석 기획	데이터 준비	데이터 분석	시스템 구현	평가 및 전개
1. 비즈니스 이해 및 범위 설정 - 구조화된 명세서 (Statement Of Work)	1. 필요 데이터 정의 및 획득방안 수립 - 계체 관계 다이어그램(Entity Relation Diagram)	1. 분석용 데이터 준비 - 비즈니스 룰 확인 및 분석용 데이터셋 추출	1. 설계 및 구현 - 아키텍쳐 및 사용자 인터페이스 설계 - 코딩 또는Business Intelligence 패키지 활용	1. 모델 발전 계획 수립 - 모델의 생명주기 설정 및 유지보수
2. 프로젝트정의 및 수행 계획 수립 - 산출물 위주로 Work Breakdown Structure 작성	2. 데이터 스토어 설계 - 관계형 데이터베이스 RDBMS	2. 텍스트 분석 - 어휘/구문 분석, 감성 분석, 토픽 분석, 오피니언 분석, 소셜 네트워크 분석(용어사전 확보)	2. 시스템 테스트 및 운영 - 단위/통합/시스템 테스트 (객관성, 완전성 확보) - 운영자, 사용자 대상 교육 실시	2. 프로젝트 평가 및 보고 - 최종보고서 작성 및 산출물, 프로세스 지적자산화 후 종료
3. 프로젝트 위험 계획 수립 - 위험 관리 계획서(회피, 전이, 완화, 수용) 작성	3. 데이터 수집 및 정합성 점검 - 크롤링, 실시간 처리, Batch 처리 등을 수집 - 메타데이터 및 데이터 사전 적용 주기적 확인	3. 탐색적 분석(EDA) - 기초 통계량, 변수 관계 활용하여 데이터 시각화(프로토타입)		
		4. 모델링 - 훈련용, 테스트용으로 데이터 분할(과적합 방지 및 일반화 이용) - 통계 모델 or 기계학습(지도학습, 비지도 학습)		
		5. 모델 평가 및 검증 - 검증용(실 운영용) 데이터		

제3절 분석 과제 발굴"상향식의 발산단계와 하향식의 수렴단계 반복 수행하여 상호 보완"

1. 하향식 접근방식(Top Down)

가. Probelm Discovery :무엇을(What) 어떤 목적으로(Why) 탐색하는지에 대한 관점, 기업 내외부 환경 포괄하는 모델, 외부 참조모델(전체적 관점의 기준 모델)

- 비즈니스 모델 기반 문제 탐색 p224

• 현재 사업+단기과제 형식 :비즈니스 모델 캔버스의 9가지 블록 단순화를 통해 업무,(재고량 최소화), 제품(기능 개선), 고객(call대기 최소화), 규제와 감사(품질 관리), 지원 인프라(적정 운영 인력)

• 새로운 문제 발굴+장기적 접근방식 : 거시적 관점의 메가트렌드(사회, 기술, 경제, 환경, 정치), 경쟁사 동향(대체재, 경쟁자, 신규 진입자), 고객 니즈(고객, 채널, 영향자=인플루언서), 역량 변화(내부 역량, 파트너와 네트워크 영역)

- 외부 참조 모델 기반 문제 탐색 : 유사 동종 환경의 사례 벤치마킹, 후보 풀, 브레인스토밍

- 분석유즈케이스 정의 : 분석 유즈케이스, 문제해결시 발생 효과 명시

나. Problem Definition :기법 정의(How) ex) '고객이탈의 증대'→이탈에 영향 미치는 요인 분석 및 이탈 가능성 예측하는 분석 문제로 변환(분석 수행 당사자 및 문제 해결시 효용을 얻는 최종 사용자 관점에서 문제 정의)

다. Solution Search

	분석역량 확보	분석역량 미확보
기존 분석기법 및 시스템	기존 시스템 개선 활용	교육 및 채용
신규 도입	시스템 고도화	전문 업체 Sourcing

라. 타당성 검토 Feasibility Study : 경제적(비용 대비 편익 분석), 데이터 및 기술적(데이터 존재 여부, 분석 시스템 환경 및 역량) → 대안 선택 후 분석과제 정의서 형태로 명시

2. 상향식 접근방식(Bottom Up)

- 기존의 하향식 접근법(Why)의 한계 극복하기 위한 방법론 : 디자인 사고 접근법(현장 관찰과 감정이입, 대상의 관점(What)으로의 전환), Unsupervised Learning 비지도학습(데이터 자체의 결합, 연관성, 유사성 등 중심으로 데이터 상태 표현_장바구니 분석, 군집분석, 기술 통계, 프로파일링 / 지도학습 : 명확한 목적으로 실시_분류, 추측, 예측, 최적화), 인과관계→상관관계

- 시행착오를 통한 문제 해결(프로토타이핑) : 분석 및 결과 확인을 통해 반복적으로 개선, 해결책이나 모형 제시(가설 생성→실험→실제 테스트→통찰→가설 확인), 동적 환경에서 신속 문제 해결

• 프로토타이핑의 필요성 :문제에 대한 인식 수준(불명확/New), 필요한 데이터 존재 불확실성, 데이터 사용 목적의 가변성(기존 데이터의 정의 재검토 후 사용 목적 및 범위 확대 가능)

3. 분석과제 정의 : 분석과제 정의서 통해 분석별 필요한 소스데이터, 분석방법, 데이터 입수 및 분석 난이도, 분석주기, 결과에 대한 검증, 상세 분석 과정 등 정의

제4절 분석 프로젝트 관리방안

5 Analytic Dimensions : Data Size, Data Complexity(정형/비정형), Speed(일/주단위, 실시간), Analytic Complexity(정확도/복잡도 Trade off 관계), Accuracy(모델과 실제값 차이 -분석의 활용성 측면)&Precision(모델 반복 수행시 편차의 수준-안정성측면)_Trade off 관계

1. 분석 프로젝트의 특성 :분석가는 정확도↑, 데이터/비즈니스 영역의 중간에서 분석 모델을 통한 조율 수행, 결과 재해석을 통한 지속적 반복, 정교화 수행하므로 프로토타이핑 방식의 어자일(Agile) 프로젝트 관리방식에 대한 고려도 필요

2. 분석 프로젝트 관리방안

관리 영역	분석 프로젝트의 특성 및 주요 관리항목
범위Scope	범위가 빈번하게 변경됨
시간	품질이 보장된다는 전데로 Time Boxing 기법으로 일정관리 진행
원가	오픈소스도구 외에 상용버전의 도구가 필요할 수 있음
품질	사전에 품질목표 수립, 품질통제와 품질보증으로 나누어 수행
통합	프로젝트관리 프로세스 통합적으로 운영
조달 Procurement	목적에 맞는 외부소싱 적절하게 운영, Poc(Proof of Concept)형태의 프로젝트는 인프라구매 아닌 클라우드 등 다양한 방안 검토
자원	전문가 확보
리스크	관련 위험 식별 및 대응방안 수립
의사소통	분석 결과 이해관계자 전체 공유
이해관계자	이해관계자의 식별 관리 필요

2장 분석 마스터 플랜

제1절 분석 마스터 플랜 수립

1. 분석 마스터 플랜 수립 프레임워크 : 정보전략계획 Information Strategy Planning 수립

우선순위 고려요소	적용 우선 순위 설정	**적용범위/방식 고려요소**	Analytics 구현 로드맵 수립
전략적 중요도		업무내재화 적용 수준	
비즈니스 성과/ROI		분석데이터 적용 수준	
실행 용이성		기술적용 수준	

2. 수행 과제 도출 및 우선순위 평가

분석 과제 도출→우선순위평가(과제 우선순위 평가기준 수립)→우선순위 정련(분석과제 수행의 선후관계 분석을 통해 순위 조정)

분석 과제 우선순위 평가기준	시급성 -전략적중요도, KPI	Value -비즈니스 효과Return)	
	난이도 -데이터 획득/저장/가공 비용, 분석 적정 지용, 분석 수준	Volume	투자비용 요소(Investment)
		Variety	
		Velocity	

난이도/시급성 매트릭스 = 포트폴리오 사분면(Quadrant) 분석
시급성 3,4,2 / 난이도 3,1,2

• 시급성↑, 난이도↓ 1사분면의 경우 경영진 및 실무자의 의사결정에 따라 난이도 조율하여 우선순위 조정 가능 ex)데이터 양 줄이거나기술적 요소(기존 시스템 미치는 영향 최소화하여 적용하거나 시스템과 별도 분리하여 수행) 조정, 분석 과제의 범위 일부로 한정 후 점차 확대 등

3. 이행계획 수립

가. 로드맵 수립 / 나. 세부 이행계획 수립 :데이터 분석 체계는 고전적인 폭포수(Waterfall) 방식 but 반복적 정련통해 완성도 높이는 방식 주로 사용, 모델링 단계는 반복적으로 수행하는 혼합형 적용

제2절 분석거버넌스 체계 수립

1. 거버넌스 체계 개요 : 데이터 분석 문화로 정착시키고 지속적으로 고도화하기 위해 조직, 프로세스, 시스템, 데이터, 교육 등으로 거버넌스 체계 구성

2. 데이터 분석 성숙도 모델 및 수준진단 : 분석준비도 6개 영역 + 분석 성숙도 3개 영역

수준진단을 통해 데이터 분석을 위한 기반, 환경 유사업종 또는 경쟁사에 비해 어느 수준인지, 데이터 활용한 분석 경쟁력 확보를 위한 보완점 등 개선 방안 도출

가. 분석 준비도

분석 업무 파악 - 발생한 사실, 예측, 시뮬레이션, 최적화 분석, 분석 업무 정기적 개선	**인력 및 조직** - 분석 전문가 교육 훈련, 관리자 기본 분석 능력, 전사 분석 총괄 조직, 경영진 분석 업무 이해력	**분석 기법** - 업무별 적합한 분석 기법 사용, 분석 업무 도입 방법론, 라이브러리, 효과성 평가, 정기적 개선
분석 데이터 - 분석 업무를 위한 데이터 충분성/신뢰성/적시성, 비구조적 데이터 관리, 외부 데이터 활용 체계, 기준데이터 관리	**분석 문화** - 사실에 근거한 의사결정, 관리자/경영진의 데이터 중시, 데이터 공유 및 협업 문화	**IT 인프라** - 운영시스템 데이터 통합, EAI, ETL 등 데이터유통체계, 분석 전용 서버 및 스토리지, 빅데이터 분석 환경

나. 분석 성숙도 모델 Capability Maturity Model Integration

단계	**도입**	**활용**	**확산**	**최적화**
설명	분석 시작/시스템 구축	결과 실제 업무에 적용	전사 차원 분석	혁신 및 성과 향상에 기여
비즈니스 부문	정기 보고 수행	예측, 시뮬레이션(운영데이터 기반)	전사 실시간 분석	외부 환경분석, 최적화
조직, 역량 부문	일부 부서, 담당자 역량에 의존	전문 담당 부서에서 관리자가 수행	전사 수행, 데이터 사이언티스트	데이터 사이언스 그룹
IT 부문	데이터 웨어하우스, 마트, ETL/EAI, OLAP	실시간 대시보드, 통계 분석 환경	시뮬레이션, 최적화, 분석 전용 서버, 비주얼 분석	협업, 내재화, 분석 Sandbox(분석실험 환경)

준비형 : 준비도↓ 성숙도↓	정착형 : 준비도↓ 성숙도↑(분석업무 제한적 사용)
도입형 : 준비도↑ 성숙도↓ 조직/인력 준비	확산형 : 준비도↑ 성숙도↑ 지속적 확산 필요

3. 분석 지원 인프라 방안 수립 : 분석 과제 단위별로 별도의 분석 시스템을 구축하는 경우 관리 복잡도 및 비용↑→분석 마스터 플랜 기획 단계에서 장기적 안정적 확장성 고려한 플랫폼 도입

- 플랫폼 :하드웨어에 탑재되어 프로그래밍 환경을 제공하는 시스템(중앙집중적, 인터페이스 최소화) → 분석 니즈 생길 때 개별 시스템 추가 X, 서비스를 추가하는 방식으로 확장성↑

• 광의의 분석 플랫폼 : 분석 서비스 제공 엔진, 분석 어플리케이션, 분석 서비스 제공 API, 운영체제 OS, **데이터처리 프레임워크, 분석 엔진, 분석 라이브러리(협의의 분석 플랫폼)**

4. 데이터 거버넌스 체계 수립

• 중복, 비표준에 따른 정합성 오류, 데이터 활용 저하 개선을 위한 전사 차원의 데이터 관리체계 필요 → 데이터 거버넌스 : 전사 차원의 모든 데이터에 대해 정책, 지침, 운영조직 등 표준화된 관리체계 수립하고 운영을 위한 프레임워크, 저장소 구축(마스터데이터, 메타데이터, 데이터사전)

• 데이터 거버넌스 체계 구축 → 데이터의 가용성, 유용성, 통합성, 보안성, 안정성 확보(빅데이터의 효율적 관리, 데이터 최적화, 정보보호, 생명주기 관리, 데이터 카테고리별 관리책임자 지정)

• 원칙(데이터 품질 유지관리 지침/가이드) + 조직(관리자 및 조직) + 프로세스(데이터 관리체계)

가. 데이터 표준화 : 데이터 표준 용어 설정, 명명 규칙 수립, 메타데이터 및 데이터사전 구축, 데이터 구축체계또는 Meta Entity Relationship Diagram 제공

나. 데이터 관리체계 :메타데이터 및 데이터사전 관리 원칙 수립 조직별 역할 및 책임, 데이터의 생명주기 관리방안 수립

다. 데이터 저장소 관리(Repository) : 전사 차원의 저장소(데이터 구조 변경에 따른 사전 영향평가 수행)

라. 표준화 활동 : 표준 준수 여부 주기적 점검 및 모니터링, 교육, 개선

5. 데이터 조직 및 인력방안 수립 (분석 조직-경쟁력 확보 위해가치 찾아최적화, Insight 전파)

- 집중형 : 별도의 독립적 분석 전담조직, 현업부서와 분석 업무가 중복 또는 이원화 될 가능성
- 기능 중심 : 해당 업무부서에서 직접 분석, 전사적 관점의 분석X, 제한적, 중복 분석
- 분산 조직 : 분석조직 인력을 현업부서에 배치, 전사 차원의 분석신속하게 실무에 적용

• 분석조직 : 비즈니스, IT기술, 분석전문 인력 **/변화관리, 교육담당 인력(겸직 가능)**

6. 분석 과제 관리 프로세스 수립

① 분석Idea 발굴 ② 분석과제 후보 제안 ③ 분석과제 확정(전사분석조직)④ 팀 구성(과제추진팀)
⑤ 분석과제 실행 ⑥ 분석과제 진행 관리(전사분석조직)⑦ 결과 공유/개선

7. 분석 교육 및 변화관리

분석내재화 단계

- 준비기 : 분석중심 문화가 미도입(균형)
- 도입기 : 기존 행태로 되돌아가려는 경향(포기多) / 성공시 강한 탄성에 의해 변화 가속화
- 안정 추진기 : 분석 활용이 일상화된 단계(균형)

• 분석기획자 -데이터분석큐레이션교육), 분석실무자-분석기법, 도구 교육, 업무 수행자 -분석기회 발굴 및 시나리오 작성법 통해 데이터 분석 및 활용이 기업 문화로 정착 확대

4과목 데이터 분석

1장 R 기초와 데이터 마트

제1절 R 기초

R의 특징 :설치용량 적음, 그래픽 처리, 데이터 처리 및 계산 능력, 패키지

R studio :R을 사용하는 통합 개발 환경 Idtegrated Development Environment

- R의 데이터 구조
• 벡터 : 논리연산자의 경우 모든 글자 대문자로 기입해야 인식, 숫자형 벡터와 문자형 벡터 합치는 경우 문자형 벡터로 전환
• 행렬 : 열을 우선 채우는 방향으로 입력(행 우선 : byrow=T)
• 데이터 프레임 : 각 열에 다른 데이터 타입 입력 가능

- 외부 데이터 불러오기 :a <- read.table("D:₩₩data₩₩example.**csv/txt**", header=T, sep=",")

```
library(RODBC)
new<- odbcConnectExcel("c:₩₩data₩₩mydata")
yourdata<- sqlFetch(new, "sheet1")    /  close(new)
```

t(a) : 전치행렬, %*% : 행렬곱, solve(a) ; 역행렬
summary(a) : 사분위수, 최소값, 최대값, 중앙값, 평균 / str(a) : 관측치, 변수개수, data 타입
a[m,n] : m번째 행, n번째 열 원소 / a[-m,] / a[, -n] : 제외 / **a[-m, -n] 불가능**

- 반복 구문과 조건문 :for (i in 1:9) { 괄호 안의 조건 하에서 / while (x<6) { 괄호 안의 조건 만족 동안 / for (i in 1:40) if (StatScore[i] >=70) over70[i] =1 else over70[i] =0 }특정 조건 만족되는 경우 이후의 구문 실행, else 이하의 조건으로 다른 조건 부여 가능

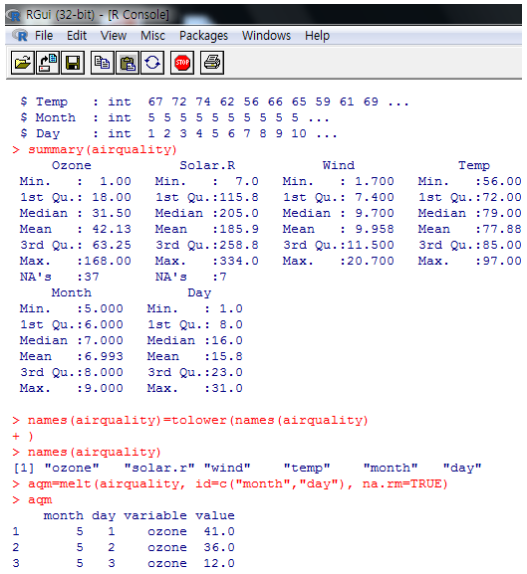
paste(number, alphabet, sep="-")
substr("BigdataAnalysis", 1, 4) → "BigD"
as.matrix 데이터형식 한가지, as.dataframe 여러 형식 가능
Sys.Date() : 현재 날짜 반환 / as.Date() : 날짜 객체 변환, format="%m/%d/%Y" mm/dd/yy
%a : 요일, %b : 월, %m : 두자리 월, %d : 두자리 일, %y : 두자리 연도, %Y : 네자리 연도

산점도 >pairs(iris[1:4], main="제목", **pch**(점모양)=21, **bh**(iris Species에 따라 다른 색상 부여)=c("red", "green3", "blue")[unclass(iris$Species)])히스토그램

히스토그램 > hist(StatScore, prob=T), > boxplot(StatScore)

제2절 데이터 마트 :데이터 웨어하우스는 정부기관 또는 정부 전체의 상세 데이터를 포함, 데이터 마트는 전체적인 데이터 웨어하우스에 있는 일부 데이터를 가지고 특정 사용자를 대상으로 한다. (사용자의 기능 및 제공 범위를 기준으로 구분)

1. R reshape(그룹화)를 활용한 데이터 마트 개발 : 데이터 정보 유지(cast, melt)



cf) 밀집화(aggregation) : 축소, 재정렬(엑셀의 피벗테이블) but 데이터 정보 손실
a<- cast(aqm, day~month~variable)

```
> a
, , variable = ozone

    month
day   5  6   7   8  9
  1  41 NA 135  39 96
  2  36 NA  49   9 78
  3  12 NA  32  16 73
  4  18 NA  NA  78 91
  5  NA NA  64  35 47
  6  28 NA  40  66 32
  7  23 29  77 122 20
  8  19 NA  97  89 23
  9   8 71  97 110 21
  10 NA 39  85  NA 24
```

d<-cast(aqm, month~variable,mean,margins=c("grand_row", "grand_col"))

```
> d
   month    ozone  solar.r      wind     temp     (all)
1      5 23.61538 181.2963 11.622581 65.54839 68.70696
2      6 29.44444 190.1667 10.266667 79.10000 87.38384
3      7 59.11538 216.4839  8.941935 83.90323 93.49748
4      8 59.96154 171.8571  8.793548 83.96774 79.71207
5      9 31.44828 167.4333 10.180000 76.90000 71.82689
6 (all) 42.12931 185.9315  9.957516 77.88235 80.05722
```

e<-cast(aqm, day~month, mean, subset=variable=="ozone")

```
> e
  day   5   6   7   8   9
1   1  41 NaN 135  39  96
2   2  36 NaN  49   9  78
3   3  12 NaN  32  16  73
4   4  18 NaN NaN  78  91
```

f <-cast(aqm, month~variable, range)

```
> f
  month ozone_X1 ozone_X2 solar.r_X1 solar.r_X2 wind_X1 wind_X2 temp_X1 temp_X2
1     5        1      115          8        334    5.7    20.1      56      81
2     6       12       71         31        332    1.7    20.7      65      93
3     7        7      135          7        314    4.1    14.9      73      92
4     8        9      168         24        273    2.3    15.5      72      97
5     9        7       96         14        259    2.8    16.6      63      93
```

X1 : min, X2 : max

2. sqldf를 이용한 데이터 분석

>sqldf("select*from iris")
>sqldf("select*from iris limit 10")
> sqldf("select count(*) from iris where Species like 'se%'") se% :Species 변수se로 시작

3. plyr

출력형태↓/입력형태→	데이터 프레임	리스트	배열
데이터 프레임	ddply	ldply	adply
리스트	dlply	llply	alply
배열	daply	laply	aaply

runif(n개 난수, 최소값, 최대값)

```
> ddply(d,"year",summarise,mean.count=mean(count))
  year mean.count
1 2012   10.50000
2 2013   11.33333
3 2014   14.16667
> ddply(d,"year",transform,mean.count=mean(count))
   year count mean.count
1  2012     5   10.50000
2  2012     7   10.50000
3  2012    11   10.50000
4  2012    18   10.50000
```

summarise 옵션 : 변수에 명령된 평균이나 합 등을 계산

transform 옵션 : summarise 옵션과 달리 계산에 사용된 변수도 출력

4. 데이터 테이블 : 빠른 그룹화, 순서화, 짧은 문장 지원, 행번호 콜론(:)으로 프린트

tables() : 현재 테이블들의 name, nrow, ncol, MB(용량), Cols(변수명), key 정보
데이터 테이블에서의 by=x 활용 시 tapply보다 빠름, x와 y 변수 모두 이용 시에는 by="x,y"

제3절 결측값 처리와 이상값 검색

1. 데이터 탐색 : cov 공분산, cor 상관계수

2. 결측값 처리NA : 결측값, NaN : 불가능한 값 Null :

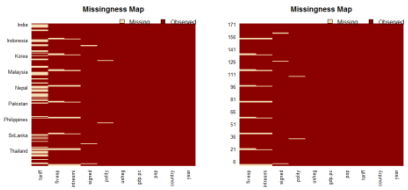
결측값 처리 패키지 : Amelia II , Mice, mistools
is.na(a) 결측값 여부 확인, a[complete.cases(a),] 결측값 포함된 관측치 삭제, !complete.cases 결측값 탐색, na.rm=T결측값 제거 옵션, na.omit(a) 결측값 제거

> a.out<-amelia(freetrade,m=5,ts="year",cs="country") m : imputation 데이터셋 개수, ts : 시계열, cs : cross-sectional 분석 정보(한 시점의 단면 데이터)

```
-- Imputation 1 --
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
-- Imputation 2 --
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
-- Imputation 3 --
  1  2  3  4  5  6  7  8  9 10 11 12 13 14
-- Imputation 4 --
  1  2  3  4  5  6  7  8  9 10 11 12 13
-- Imputation 5 --
  1  2  3  4  5  6  7  8  9 10 11 12 13
```

freedata <- a.out$imputations[[5]]$**tariff** 보정

missmap(a.out) missmap(freedata)



3. 이상값 검색 :전처리 방법 결정, 부정사용방지시스템(Fraud Detection System) 규칙 발견 시

① 의도하지 않게 잘못 입력된 경우 : bad data
② 의도하지 않게 입력됐으나 분석 목적에 부합하지 않아 제거해야 하는 경우 : bad data
③ 의도되지 않은 현상이지만 분석에 포함해야하는 경우 :**이상값**
④ 의도된 **이상값** → 사기(fraud)

일정시간에 일괄적으로 거래정보 입력한 경우 시간대별매출 분석에서는 제외(②), 고객 행동 분석에서는 포함(③)

관련 알고리즘 : Extreme Studentized Deviation, MADM /**3*표편** 떨어진 경우 이상값 판단

2장 통계 분석

제1절 통계학 개론

가. 통계학 : 유용한 정보 이끌어내는 작업, 자료 수집 및 정리, 해석하는 방법 모두 포함
나. 모집단과 표본 : 모집단을 구성하는 개체 추출단위 또는 원소, 총조사/표본조사
다. 표본추출의 방법 : 단순랜덤추출법/계통추출법/집락추출법/층화추출법
- 표본조사 : 대상 집단의 일부를 추출해 관측, 조사
- 실험 : 특정 목적 하 실험 대상에게 처리 가한 후 결과 관측
라. 자료의 종류
- 명목척도 : 성별, 출생지 등 어떤 집단에 속하는지 분류
- 순서척도 : 측정 대상의 서열관계 관측, 선호도
- 구간척도 : 측정 대상의 속성 양을 측정, 온도, 지수, 절대적 원점 없음
- 비율척도 : 절대적 기준인 0값 존재, 무게, 나이, 연간 소득, 제품가격

2. 통계 분석

통계적 추론 : 수집된 자료를 이용해 대상 집단(모집단)에 대한 의사결정
- 추정 : 대상 집단의 특성값(모수) 추측
- 가설검정 : 대상 집단에 대해 특정한 가설을 설정한 후에 그 가설의 채택 여부 결정
- 예측 : 미래의 불확실성을 해결해 효율적인 의사결정을 위해 수행

3. 확률 및 확률분포

• 근원사건 : 한개의 원소로만 이루어진 사건
• 배반사건 : 교집합 공집합
• P(B|A)=P(A∩B)/P(A), 만약 =P(B) 라면 두 사건 A, B 독립
• 확률변수 : 정의역이 표본공간이고 치역이 실수값인 함수(이산형 분포-베르누이, 이항, 기하, 다항, 포아송 / 연속형 분포-균일, 정규, 지수, t, 카이제곱, F)
• 가설검정 :귀무가설(H0, 대립가설과 반대), 대립가설(H1, 뚜렷한 증거가 있어야 채택하는 가설), p-value(H0 사실일 때 관측된 검정통계량 값보다 H1 지지하는 검정통계량이 나올 확률) 유의수준보다 낮으면 H0 기각, 기각역(H0 기각하는 통계량의 영역)

정확한 사실	H0 사실이라고 판정	H0 사실 아니라고 판정
H0 사실	옳은 결정	1종 오류(알파)
H0 사실 아님	2종 오류(베타)	옳은 결정

알파와 베타는 상충관계, 알파의 크기를 0.05 등으로 고정 후 베타가 최소가 되도록 기각역 설정

4. 비모수 검정

- 모수적 검정방법 : 모집단의 분포에 대한 가정 후, 그 가정 하에 검정통계량과 분포 검정 실시

- 비모수적 검정방법 : 자료가 추출된 모지단의 분포에 대해 아무 제약 없이 검정 실시(분포의 형태가 동일하다, 동일하지 않다 가설 설정, 관측값의 절대적 크기에 의존하지 않는 관측값들의 순위나 두 관측값 차이의 부호를 이용해 검정-부호검정, 윌콕슨 순위합검정/부호순위합검정, 만-위트니의 U검정, 런검정, 스피어만의 순위상관계수

제2절 기초 통계 분석

- 회귀분석 :회귀계수 추정량=최소제곱추정량(Least Squares Estimator)
• 모형 통계적 유의미 : F통계량 pvalue
• 회귀계수 유의미 : 해당 계수의 t통계량, pvalue, 신뢰구간
• 모형 설명력 : 결정계수
• 데이터 적합 여부 : 잔차 회귀진단
• 가정 : 선형성, 독립성(잔차/독립변수), 등분산성(독립변수 오차분산 일정), 비상관성(관측치들의 잔차들끼리), 정상성(잔차항 정규분포)
• 설명변수 선택 : 모든 가능 조합 회귀분석, 전진선택(forward), 후진제거(backward), 단계별(stepwise)
• step(**lm**(종속변수~설명변수, 데이터셋), scope=list(lower=~1,upper=~설명변수), direction="변수선택방법")후진 : 설명변수., scope 생략) / 단계적, 전진 : 설명변수 1

제3절 다변량 분석

1. 상관분석 산점도(plot) 직선이면 상관계수 r=0(독립)

- 피어슨 상관계수 :rcorr(as.matrix(mtcars), type="pearson") → 상관계수, pvalue 출력
- 스피어만 상관계수 :두 변수 간의 비선형적인 관계도 나타낼 수 있는 값, type="spearman"

2. 다차원척도법(MultiDimensional Scaling)**cmdscale**
: 여러 대상 간의 거리가 주어져 있을때 대상들을 상대적 거리를 가진 실수공간의 점들로 배치

3. 주성분분석 (Principal Component Analysis)

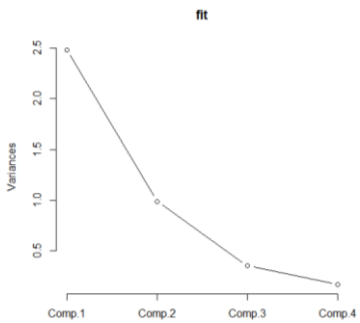
: 상관관계가 있는 고차원 자료를 자료의 변동을 최대한 보존하는 저차원 자료로 변환(축약)

분산이 가장 큰 선형변환을 첫번째 주성분($Y_{i=a_i^T X}$)이라고 하며, 정의에 따라 주성분들은 서로 상관관계가 없고 주성분들의 합은 변수들의 분산의 합과 같다.(i번째 주성분의 로딩, a_i)

```
> fit<-princomp(USArrests, cor=TRUE)
> summary(fit)
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4
Standard deviation     1.5748783 0.9948694 0.5971291 0.41644938
Proportion of Variance 0.6200604 0.2474413 0.0891408 0.04335752
Cumulative Proportion  0.6200604 0.8675017 0.9566425 1.00000000
> loadings(fit)

Loadings:
         Comp.1 Comp.2 Comp.3 Comp.4
Murder   -0.536  0.418 -0.341  0.649
Assault  -0.583  0.188 -0.268 -0.743
UrbanPop -0.278 -0.873 -0.378  0.134
Rape     -0.543 -0.167  0.818

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
> plot(fit,type="lines")
```



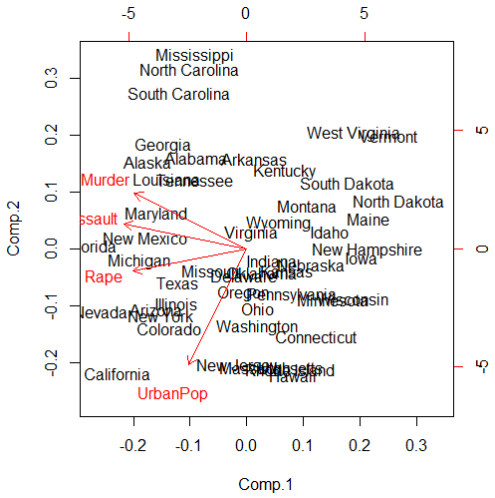
cor=T 상관계수 행렬 사용하여 주성분분석 수행
scale=T 표준화

→ 첫번째 주성분이 전체 분산의 62% 설명, 두번째, 세번째 주성분은 전체 분산 중 25%, 9% 설명

$Y_1 = -0.536\text{Mrder} - 0.583\text{Assault} - 0.278\text{UrbanPop} - 0.543\text{Rape}$
$Y_2 = 0.418\text{Mrder} + 0.188\text{Assault} - 0.873\text{UrbanPop} - 0.167\text{Rape}$

→ 스크리그림 통해 주성분의 분산이 급격하게 감소하여 주성분의 개수를 늘릴때 얻는 정보의 양이 미미한 지점에서 주성분의 개수 결정 or 주성분들이 설명하는 분산 비율 70~90%일때 결정

fit$scores : 각 관측치를 주선분들로 표현한 값 나타냄
biplot(fit) : 관측치들을 첫번째와 두번째 주성분의 좌표에 그린 그림



첫번째 주성분 : Assault, Murder, Rape와 비슷한 방향을 가지고 UrbanPop 방향과 수직에 가까운 것으로 보아 Assault, Murder, Rape 변수들에 대해 상대적으로 큰 가중치 적용하여 계산

두번째 주성분 : UrbanPop 상대적으로 평행하기때문에 다른 변수들에 비해 UrbanPop의 영향을 받아 구성된 것

→ 첫번째 주성분의 값이 작을수록 세가지 범죄 발생율이 큰 주, 두번째 주성분 값이 작을수록 도심인구 비율이 큰 주라고 해석

http://blog.naver.com/skkong89?Redirect=Log&logNo=90117511177

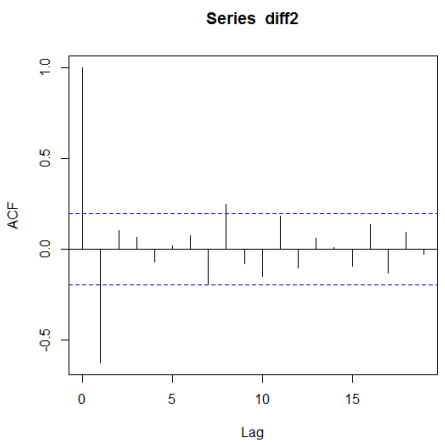
제4절 시계열 예측

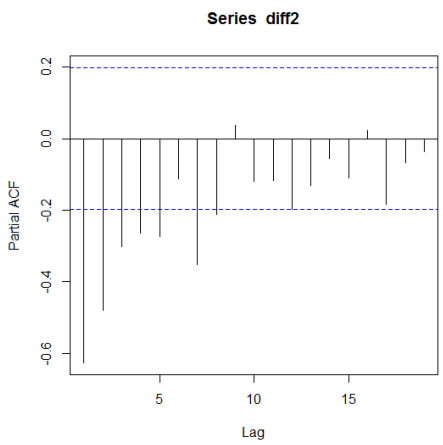
1. 정상성 : 평균 일정, 분산이 시점에 미의존, 공분산은 시차에 의존(시점 자체에 의존X)

2. 시계열 모형
- 자기회귀모형(AR(p)모형) :현 시점의 자료가 p시점 전의 유한 개의 과거 자료로 설명
• $Z_t = \emptyset_1 Z_{t-1} + \emptyset_2 Z_{t-2} + \cdots + \emptyset_p Z_{t-p} + a_t$($a_t$: 평균 0, 독립, 백색잡음과정(정상시계열))
• 자기상관함수(ACF) 시차가 증가함에 따라 점차 감소, 부분자기상관함수(PACF) p+1 시차 이후 급격하게 감소하여 절단된 상태
- 이동평균모형(MA(p)모형) :현 시점의 자료를 유한개의 백색잡음의 선형결합으로 표현(정상성)
• $Z_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_p a_{t-p}$
• ACF p+1 시차 이후 절단된 형태, PACF 점차 감소하는 형태
- 자기회귀누적이동평균모형(ARIMA(p,c,q) : 비정상 시계열 모형(d=0이면 ARMA(p,q)모형-정상성, p=0이면 IMA(d,q)모형이며 d번 차분하면 MA(q)모형/q=0이면 ARI(p,d)모형이며 d번 차분하면 AR(p)모형)
• 분해 시계열 : 추세요인/계절요인(고정적 주기)/순환요인(알려지지않은 주기)/불규칙요인(오차)-

decompose 함수통해 분해 가능

• 차분 :diff2<-diff(Nile,differences=2), acf(diff2), pacf(diff2, plot=FALSE)





ARMA(8,0) : PACF lag=0에서 절단

ARMA(0,1) : ACF lag=2에서 절단

• 예측 : Nile.arima<-arema(Nile,order=c(1,1,1)) / Nile.forecast<-**forecast**(Nile.arima,**h=10**)

3장 정형 데이터 마이닝

데이터 마이닝 : 일일거래 데이터, 고객 반응 데이터 등 모든 사용가능한 원천 데이터를 기반으로 감춰진 지식 발견, 기대하지 못했던 경향, 규칙 발견하여 이를 실제 비즈니스 의사결정 등에 활용

제1절 데이터 마이닝 개요

1. 분류 : 새롭게 나타난 현상을 검토하여 기존의 분류, 정의된 집합에 배정(범주화), 의사결정나무, memory-based reasoning, link analysis,
2. 추정 : 주어진 연속 입력 데이터를 사용하여 알려지지 않은 결과 값 추정, 신경망 모형
3. 예측 : 미래의 값 추정하는 것 제외하면 분류나 추정과 동일, 장바구니 분석, momory-based resoning, 의사결정나무, 신경망
4. 연관 분석 :같이 팔리는 물건과 같이 아이템의 연관성 파악, 장바구니 분석
5. 군집 : 이질적인 모집단을 동질성을 지닌 그룹별로 세분화, 선분류된 기준에 미의존, 데이터 마이닝이나 모델링의 준비단계로 사용
6. 기술 : 상품에 관한 이해 증가시키는 것 description

목적 정의(이해관계자가 모두 동의, 1단계부터 전문가 참여하여 데이터 정의)→데이터 준비(데이터 정제 및 보강)→데이터 가공(목적변수 정의 및 데이터 형식 가공)→데이터 마이닝 기법 적용(데이터 마이닝 기법 1단계 목적 정의 시 결정)→검증(자동화 방안, 테스트 마케팅이나 과거 데이터 활용 가능-테스트 마케팅과 모델링 차이 구분 ex) 휴면고객 재탈환 가능성 모델링, 휴면 고객을 대상으로 재탈환 캠페인을 펼쳤을 때 반응할 고객 모델링

제2절 분류 분석 : 반응변수가 범주형인 경우 분류, 연속형인 경우 예측하는 것이 목적

1. 로지스틱 회귀모형 : 반응변수 범주형인 경우, 모형 적합 통해 추정된 확률 사후확률

> a<-subset(iris,Species=="setosa"|Species=="versicolor")
>a$Species<-factor(a$Species)

```
> str(a)
'data.frame':   100 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 2 levels "setosa","versicolor": 1 1 1 1 1 1 1 1 1 1 ...
```

>b<-glm(Species~Sepal.Length, data=a,family=binomial)>summary(b)

```
Call:
glm(formula = Species ~ Sepal.Length, family = binomial, data = a)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.05501  -0.47395  -0.02829   0.39788   2.32915

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -27.831      5.434  -5.122 3.02e-07 ***
Sepal.Length    5.140      1.007   5.107 3.28e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 138.629  on 99  degrees of freedom
Residual deviance:  64.211  on 98  degrees of freedom
AIC: 68.211

Number of Fisher Scoring iterations: 6
```

Sepal.Length가 한단위 증가하면 Versicolor 오즈가 exp(5,140) 약 170배 증가

Residual deviance는 예측변수 Sepal.Length가 추가된 적합모형의 이탈도

Null deviance에 비해 자유도 1기준에 이탈도의 감소가 74.4 정도 감소

2. 신경망 모형 : 인공신경망에서 입력은 시냅스에 해당하며 개별신호의 강도에 따라 가중되며, 활성함수는 인공신경망의 출력을 계산한다. 많은 데이터에 대한 학습을 거쳐 원하는 결과가 나오도록(오차↓) 가중치 조정된다.

- 가중치는 의사결정 경계의 방향을 나타내는 모수, 편의는 의사결정 경계의 위치를 결정하는 모수, 가중치와 절편은 학습을 통해 오차제곱합이 최소가 되는 방향으로 갱신

• 부호(sign) or threshold 함수 : 결과 이진형 -1, 0

• 계단(step) 함수 :결과 이진형 0, 1

• 시그모이드(sigmoid) 함수 : 결과 연속형, 0<=y<=1$y = \frac{1}{1+\exp(-z)}$

• Softmax 함수 :표준화지수(일반화로지스틱) 함수, 출력값이 여러 개로 주어지고, 목표치가 다범주인 경우 각 범주에 속할 사후 확률 제공$y = \frac{\exp(z_j)}{\sum_{i=1}^{L}\exp(z_i)}$

• tanh 함수 : 결과 연속형, -1<=y<=1 $y = \frac{\exp(z)=\exp(-z)}{\exp(z)+\exp(-z)}$

• 가우스(Gauss) 함수 : 결과 연속형, 0<=y<=1$y = \exp(-\frac{z^2}{2})$

>**nnet**(Species~., data=iris, size=2,rang=0.1,decay=5e-4,maxit=200) 연결선의 가중치=굵기

> net.infert<-neuralnet(case~age+parity+induced+spontaneous, data=infert,hidden=2,err.fct="ce", linear.output=FALSE, likelihood=TRUE)

$generalized.weights가 제시하는 일반화 가중치는 각 공변량들의 효과를 나타낸 것으로 로지스틱 회귀모형에서의 회귀계수와 유사하게 해석(각 공변량이 로그-오즈에 미치는 기여도), 작은 분산은 선형효과 제시, 큰 분산은 관측치 공간 상에서 변화가 심하다는 것 즉 비선형적 효과 제시

• 입력층 : 자료벡터, 패턴을 받아들임 / 은닉층 : 이전층으로부터 출력받아 가중하여 비선형의 활성함수로 넘김 / 출력층 : 최종은닉층으로부터 결과 받아 비선형적으로 결과 넘겨 목표값 제공

• 다층신경망은 단층신경망에 비해 훈련이 어렵다. / 시그모이드 활성함수를 가지는 2개 층의 네

트워크(1개 은닉층)는 임의의 의사결정 경계를 모형화할 수 있다.
• 출력 층 노드의 수는 출력 범주의 수로 결정, 입력의 수는 입력 차원의 수로 결정, 은닉층 노드의 수 너무 적으면 네트워크가 복잡한 의사결정 경계 만들수 없으며 너무 많으면 네트워크 일반화 어렵다.

3. 의사결정나무 모형 : 상위노드에서의 분류변수, 분류기준값은 이 기준에 의해 분기되는 하위노드에서 노드(집단) 내 동질성, 노드 간 이질성 커지도록 선택
• 뿌리 마디(맨 위의 마디), 부모마디(상위 마디), 자식마디(하위 마디), 최종 마디(더이상 분기되지 않는 마디), 가지 분할(나무가지 생성하는 과정), 가지치기(생성된 가지 잘라내어 모형 단순화)
• 분류나무(목표변수가 이산형인 경우), 회귀나무(목표변수가 연속형인 경우)
• 분류나무 : 상위 노드에서 가지 분할 시 분류(기준)변수와 분류기준값의 선택방법으로 카이제곱 통계량의 pvalue(p값이 작을수록 자식 노드 내 불확실성=이질성↑), 지니계수(클수록 이질적, 순수도↓) 엔트로피 지수 사용- 이 값들 작아지는 방향으로 가지 분할 수행

알고리즘	이산형 목표변수	연속형 목표변수
CHAID 다지분할	카이제곱 통계량	아노바 F통계량
CART 이진분할	지니계수	분산감소량
C4.5	엔트로피지수	-

• 의사결정나무 분석과정 : 목표변수와 관계 있는 설명변수 선택→분석목적과 자료구조에 따라 적절한 분리기준, 정지규칙 정하여 의사결정나무 생성→부적절한 나무가지 제거(가지치기)→이익, 위험, 비용 고려하여 모형 평가→분류 및 예측
• **rpart** 패키지/함수, rpart.plot 패키지의 prp 함수, party 패키지의 ctree함수,
• 의사결정나무 모형의 장점 : 구조 단순하여 해석이 용이, 유용한 입력변수 파악, 예측변수간 상호작 용 및 비선형성 고려하여 분석 수행, 수학적 가정 불필요 / 단점 : 분류기준값 경계선 근방의 자료값에 대해서는 오차가 큼(비연속성), 로지스틱회귀와 같이 각 예측변수의 효과를 파악하기 어려우며, 새로운 자료에 대한 예측 불안정

4. 앙상블 모형 : 여러 분류모형에 의한 결과를 종합하여 분류의 정확도 높이는 방법, 적절한 표본추출법으로 데이터에서 여러 개의 훈련용 데이터 집합을 만들어 각각의 데이터 집합에서 하나의 분류기를 만들어 앙상블

가. 배깅(**bagging**) : bootstrap aggregating, 원데이터 집합으로부터 크기 같은 표본을 여러번 단순임의복원추출하여 각 (붓스트랩)표본에 대해 분류기(classifiers)를 생성한 후 결과 앙상블

나. 부스팅(**boosting**) :붓스트랩 표본을 구성하는 재표본 과정에서 각 자료에 동일한 확률을 부여하지 않고 분류가 잘못된 데이터에 더 큰 가중을 주어 표본 추출, 붓스트랩 표본을 추출하여 분류기를 만든 후 분류결과를 이용하여 각 데이터가 추출될 확률 조정 후 다음 붓스트랩 표본 추출하는 과정 반복(아다부스팅 : 가장 多 사용 부스팅 알고리즘)

```
>pred<-predict(boo.adabag, newdata=iris)
>tb<-table(pred$class, iris[ ,5])
>error.rpart<- 1-(sum(diag(tb)/sum(tb)) : 오분류율 계산
>iris[iris$Species!="setosa", ] ->iris
>n<-dim(iris)[1]
>trind<-sample(1:n, floor(.6*n), FALSE)훈련용/검증용 자료 구분
>teind<-setdiff(1:n,trind)
>iris[ ,5]<-as.factor((levels(iris[ ,5])[2:3])[as.numeric(iris[ ,5])-1])
>gdis<-ada(Species~., data=iris[trind, ], iter=20, nu=1, type="discrete")
>gdis<-addtest(gdis, iris[teind, -5], iris[teind, 5])
>gdis  >plot(gdis, TRUE, TRUE)카파계수 훈련용, 검증용 자료 그림
```

다. 랜덤포레스트(**randomForest**) :배깅+랜덤과정(원자료로부터 붓스트랩 샘플 추출하여 각 붓스트랩 샘플에 대해 트리 형성, 각 노드마다 모든 예측변수 안에서 최적의 분할을 선택하는 방법 대신 예측변수들 임의로 추출하고 추출된 변수 내에서 최적의 분할을 만들어 나가는 방법)
• proximity=TRUE :오류율에 대한 OOB(out-of-bag) 추정치 제공 옵션
• improtance(), varImpPlot() : 해당 변수로부터 분할이 일어날 때 불순도(Impurity)의 감소가 얼마나 일어나는지 나타내는 값(불순도↓→순수도↑), 지니 지수는 노드의 불순도 나타내는 값
• 마진(margin) : 랜덤포레스트의 분류기 가운데 정분류를 수행한 비율에서 다른 클래스로 분류한 비율의 최대치를 뺀 값, 양의 마진은 정확한 분류 의미

5. 모형 평가

- 모형 평가의 기준
• 일반화 가능성 : 같은 모집단 내 타 데이터에 적용하는 경우 안정적 결과 제공 의미(확장)
• 효율성 : 적은 입력변수를 필요로 할수록 효율성↑
• 예측과 분류의 정확성 : 구축된 모형의 정확성 측면에서 평가
- 분류 분석 모형 평가를 위해 로데이터에서 모형 구축을 위한 훈련용 자료와 모형 성과 검증을 위한 검증용 자료 추출 : 주어진 데이터에서만 높은 성과를 보이는 과적합화(overfitting) 해결(잘못된 가설 가정하는 2종 오류 발생 방지)
• 홀드아웃 방법 :원천 데이터를 랜덤하게 두 분류(훈련용 70%, 검증용 30%)로 분리하여 교차검정 실시
• 교차검증 : 반복적으로 성과 측정하여 그 결과 평균한 것으로 분류 분석 모형 평가(k-fold 교차검증 : 전체 데이터를 사이즈 동일한 k개의 하부 집합으로 나누어 k번째의 하부 집합을 검증용 자료로, 나머지 k-1개의 하부 집합을 훈련용 자료로 사용, 이를 k번 반복 측정하여 각각의 반복 측정 결과를 평균 낸 값을 최종 평가로 사용)

```
>set.seed(1234) >k=10 >iris<-iris[sample(nrow(iris)), ]
>folds<-cut(seq(1,nrow(iris)), breaks=k, labels=FALSE)  >traindata=list(0) >testdata=list(0)
>for (i in 1:k) { testIdx<-which(folds=i, arr.ind=TRUE)testdata[[i]]<-iris[testidx, ]
traindata[[i]]<-iris[-testidx, ] }
>head(traindata[[1]]) >head(testdata[[2]])
```

• 붓스트랩 방법 :평가를 반복하는 측면은 교차검증과 유사하나, 훈련용 자료 반복 재선정 차이점, 관측치를 한번 이상 훈련용 자료로 사용하는 복원추출법에 기반, 관측치 d개일 경우, (1-1/d)*d =>0.368 즉, 36.8% 관측치는 훈련용 집합으로 선정되지 않으며 나머지 63.2% 훈련용

가. 오분류표(**confusion matrix**)

		예측치		
		TRUE (Positive)	FALSE (Negative)	합계
실제값	TRUE	TP	FN	P
	FALSE	FP	TN	N
	합계	P;	N;	P+N

• TP(True Positives) : 실제값과 예측치 모두 True인 빈도
• TN(True Negatives) : 실제값과 예측치 모두 False인 빈도
• FP(False Positives) : 실제값 False지만 True로 예측한 빈도
• FN(False Negatives) : 실제값 True지만 False로 예측한 빈도
• 정분류율(accuracy, recognition rate) : 전체 관측치 중 실제값과 예측치 일치한 정도(범주의 분포가 균형을 이룰 때 효과적인 평가지표) (TP+TN) / (P+N)
• 오분류율(error rate, misclassifcation rate) : 전체 관측치 중 실제값과 예측치 다른 정도 , 1-accuracy, (FP+FN) / (P+N)
• 분류 분석 대상 대부분은 소수 집단에 대한 예측 실패 비용이 다수 집단에 대한 예측 실패 비용보다 크다. ex) 암환자 분류
• 범주 불균형 문제를 가지고 있는 데이터에 대한 분류 분석 모형의 평가 지표 :민감도(sensitivity, 실제값이 True인 관측치 중 예측치가 적중한 정도, TP/T), 특이도(specificity, 실제값이 False인 관측치 중 예측치가 적중한 정도, TN/N)
• 정확도(precision) : True로 예측한 관측치 중 실제값이 True인 정도 TP/(TP+FP)
• 재현율(recall) : 실제값이 True인 관측치 중 예측치가 적중한 정도로 민감도와 동일(완전성을 평가하는 지표) TP/(TP+FN) = TP/P
• F1지표(F1 score) : 정확도와 재현율 상반 관계이므로 이를정확도와 재현율에 같은 가중치부여하여 조화평균 보정$\beta = 1, F_\beta = \frac{(1+\beta^2)*\text{Precision}*\text{Recall}}{\beta^2*\text{Precision}+\text{Recall}}$(F2는 재현율 가중치 정확도의 2배)

나. ROC(Receiver Operating Characteristic) 그래프 : 두 분류 분석 모형 비교 결과 시각화, x축에 FP Ratio(1-특이도), y축에는 민감도, ROC 그래프 밑부분 면적(Area Under the Curve) 넓을수록 좋은 모형으로 평가(AUC 1에 가까울수록 좋은 모형)

다. 이익도표와 향상도 곡선

• 이익(gain) : 목표 범주에 속하는 개체들이 각 등급에 얼마나 분포하는지 나타내는 값, 해당 등급에 따라 계산된 이익값 누적으로 연결한 도표가 이익 도표(분류 분석 모형을 사용하여 분류된 관측치가 각 등급별로 얼마나 포함되는지 나타내는 도표)
• 향상도 곡선(lift curve) : 랜덤 모델과 비교하여 해당 모델의 성과가 얼마나 향상되었는지 등급별로 팡가하는 그래프(하위 등급으로 갈수록 향상도 감소하는 것이 모형의 예측력 적절함 의미하며, 등급에 관계없이 향상도 차이 없으면 예측력 좋지 않음 의미)- ROCR 패키지performance 함수
* 향상도 곡선 해석 : rate of positive predictions 0.2, lift value 2 → 상위 20% 집단에 대해 랜랜덤모델 비교할 때 신경망 모형이 약 2배 성과 향상을 보인다.

제3절 군집 분석 : n개의 개체들을 유사한 성격을 가지는 몇 개의 군집으로 집단화하고 군집들 사이의 관계를 분석하는 다변량분석 기법, 별도의 반응변수 요구되지 않으며 오로지 개체들 간의 유사성에만 기초하여 군집 형성, 이상값 탐지에도 사용

1. 계층적 군집(hierarchical clustering) : 가장 유사한 개체를 묶어 나가는 과정 반복하여 원하는 개수의 군집 형성, 계통도 또는 덴드로그램의 형태로 결과 주어지며 각 개체는 하나의 군집에만 속하게 됨
- 병합적(agglomerative) 방법 : 작은 군집부터 출발하여 군집을 병합, 매 단계마다 모든 그룹 쌍 간의 거리를 계산하여 가까운 순으로 병합 수행하여 한 개 그룹만 남거나 종료 조건조 될 때까지 반복(hclust, agnes, mclust 함수)
- 분할적(divisive) 방법 : 큰 군집으로부터 출발하여 군집 분리(diana, mona 함수)
- 군집 간의 거리 측정하는 방법(p511)
• 최단/단일연결법(single linkage method) : 두 군집 사이의 거리를 각 군집에서 하나씩 관측값을 뽑았을 때 나타날 수 있는 거리의 **최소값**으로 측정, 고립된 군집을 찾는데 중점
• 최장/완전연결법(compete ~) :최단/단일연결법 반대(**최대값**), 군집들의 내부 응집성에 중점
• 중심연결법(centroid ~) :두 군집의 중심 간 거리 측정, 두 군집이 결합될 때 새로운 군집의 평균은 가중평균
• 평균연결법(average ~) :모든 항목에 대한 거리 평균 구하면서 군집화
• 와드연결법(ward ~) :군집내의 오차제곱합에 기초하여 군집 수행, 두 군집이 합해지면 병합된 군집의 오차제곱합이 병합 이전보다 커지게 되는데, 그 증가량이 가장 작아지는 방향으로 군집 형성해나가는 방법(크기가 비슷한 군집끼리 병합하는 경향)
- 개체 간의 유사성(거리) 정의(유클리드, 맨하튼, 민코우스키 : 수학적 거리 / 표준화(측정 단위), 마할라노비스(측정 단위+변수간 상관성 고려) : 통계적 거리)
• 유클리드(Euclidian) 거리 :$d(i,j) = \sqrt{\sum_{f=1}^{p}(x_{if} - x_{jf})^2}$
• 맨하튼(Manhattan) 또는 시가(city-block) 거리 :$d(i,j) = \sum_{f=1}^{p}|x_{if} - x_{jf}|$
• 민코우스키(Minkowski) 거리 :$d(i,j) = [\sum_{f=1}^{p}(x_{if} - x_{jf})^m]^{1/m}$
• 표준화(standardized) 거리 :$d(i,j) = \sqrt{(x_i - x_j)' D^{-1}(x_i - x_j)}$, D : 표본분산 대각행렬
• 마할라노비스(Mahalanobis) 거리 :$d(i,j) = \sqrt{(x_i - x_j)' S^{-1}(x_i - x_j)}$, S : 표본 공분산 행렬
• 체비셰프(Chebychev), 켄버라(Canberra) 거리도 있으며, 유사성 측도인 코사인 거리, 상관계수 등을 이용하여 거리 정의 가능,
• 모든 변수 명목형인 경우 $d(i,j)$=개체 i와 j에서 다른 값을 가지는 변수의 수/총변수의 수
• 단순 일치 계수(명목형 자료에 대한 거리), 순위상관계수(순서형 자료)
• cutree() : 계층적 군집 결과를 tree높이(h), 그룹 수(k) 옵션 지정 가능
• rect.hclust() :덴드로그램
>agnes(USArrests, metric="manhattan", stand=TRUE)method="flexible"/daisy= 옵션 가능

2. k-평균 군집(장점 : 알고리즘 단순, 빠르며, 많은 데이터 가능, 모든 변수 연속적인 경우 / 단점 : 잡음이나 이상값 영향 大, 볼록한 형태 아닌 군집(U형태)이 존재하면 성능 저하)
①초기 군집 중심으로 k개의 객체 임의 선택 ②각 자료를 가장 가까운 군집 중심에 할당 ③각 군집 내의 자료들의 평균 계산하여 군집 중심 갱신 ④군집 중심 변화 없을 때 중지
→군집의 매 단계마다 군집 중심으로부터의 오차제곱합 최소화 하는 방향으로 군집 형성해 나가는 탐욕적 알고리즘으로 간주, 안정된 군집 보장 but 최적 X, 부분 최적화
• 군집의 수 k는 미리 임의로 결정하여 k 중심값 임의로 선택(초기 중심점들은 서로 멀리 떨어져

있는 것이 바람직하며, 초기값에 따라 군집 결과 좌우
→ kmeans 함수의 nstart 옵션 다중 초기값에 대한 최적의 결과 제시
>nc<-Nbclust(df, min.nc=2,max.nc=15,method="kmeans")>table(nc$Best.n[1,]) : 군집수 결정
• 이상값 민감한 단점 보완 : k-중앙값 군집(pam함수), k평균군집 수행 전 이상값 제거
• randIndex 함수 : 실제 와인의 종류와 군집간의 일치도 나타내는 수정 순위지수(adjusted rank index : 우연에 의해 발생되는 경우 고려한 값으로 -1과 1사이의 값)
• kcca 함수 : k중심군집 수행 family= 옵션 "kmeans", "kmedians", "angle", "jaccard", "ejaccard"
• cclust 함수 : convex clustering 수행 method= 옵션 "kmeans", "hardcl", "neuralgas"

3. 혼합분포군집 :k개의 모수적 모형의 가중합으로 표현되는 모집단 모형으로부터 데이터가 나왔다는 가정 하에 모수와 함께 가중치를 자료로부터 추정하는 모형 기반의 군집 방법
- 각 데이터가 어느 집단으로부터 나온 건지 모르므로 이에 대한 정보를 가지는 잠재변수(latent variable) 도입
• E단계 :임의의 파라미터 값 결정, 잠재변수 Z의 기대치 계산
• M단계 : 잠재변수 Z의 기대치를 이용하여 파라미터 추정하여 likelihood가 최대치면 추정값 도출, 아니면 Z 기대치 계산
- 혼합분포군집은 k평균군집의 절차와 유사하나 확률분포를 도입하여 군집을 수행하는 군집 방법으로 군집을 몇 개의 모수로 표현할 수 있으며, 서로 다른 크기나 모양의 군집 찾을 수 있다.
- EM알고리즘을 이용한 모수 추정에서 데이터가 커지면 수렴하는데 시간이 걸리며, 군집 크기가 너무 작으면 추정의 정도가 떨어지거나 어려우며 이상값에 민감
4. SOM(Self-Organizing Maps 자기조직화지도) = 코호넨 맵(Kohonen Maps): 비지도 신경만으로 고차원의 데이터를 이해하기 쉽게 저차원의 뉴런으로 정렬하여 지도의 형태로 형상화
• 입력 변수의 위치 관계를 그대로 보존
• 입력층 : 입력벡터 받는 층, 입력 변수의 개수와 동일하게 뉴런 수 존재, 입력층의 자료는 학습을 통하여 경쟁층에 정렬(지도)
• 경쟁층 : 입력벡터의 특성에 따라 벡터가 한 점으로 클러스터링 되는 층, 군집 수만큼 뉴런 존재
① SOM 맵의 뉴런(노드)에 대한 연결 강도 초기화 ② 입력벡터 제시 ③ 유클리드 거리를 사용하여 입력벡터와 프로토타입벡터(경쟁층의 각각의 뉴런) 사이의 유사도 계산 ④ 입력벡터와 가장 거리가 짧은 프로터타입벡터(Best Matching Unit) 탐색 ⑤BMU와 그 이웃들 연결 강도 재조정 ② 돌아가서 반복 → 연결강도는 입력 패턴과 가장 유사한 경쟁층 뉴런이 승자가 되며 경쟁층에는 승자 뉴런만 나타남(승자 독식 구조)
• SOM 이용한 군집분석은 역전파(back propagation) 알고리즘을 이용하는 인공신경망과 달리 하나의 전방 패스(feed=forward flow)를 사용함으로써 속도 빠름(실시간 학습 처리 가능)

4절 연관 분석

1. 연관규칙

가. 연관규칙의 개념 : 항목들 간의 조건-결과 식으로 표현되는 유용한 패턴
• 트랜잭션(transaction) : 특정 고객, 즉 장바구니 하나에 해당하는 정보
• 연관성 규칙의 일반적인 형태 조건과 반응(if-then) but 모든 규칙이 유용하지 않을 수 있음

나. 연관규칙의 측정지표
- 지지도(support) : 전체 거래 중에서 품목 A, B가 동시에 포함되는 거래의 비율, P(A∩B)
- 신뢰도(confidence) : 품목 A가 포함된 거래 중 품목 A, B 동시에 포함하는 확률, P(A∩B)/P(A)
- 향상도(lift) : 품목 B 구매한 고객 대비 품목 A 구매한 후 품목 B 구매하는 고객 확률, P(A|B)/P(B) → 향상도 1이면 A, B 거래 독립이고 1보다 크면 A, B 거래 연관성 높음 의미
• 최소 지지도 정해 규칙 도출(지지도를 높은 값에서 낮은 값으로 변경해가며 실행)

다. 연관 분석 절차

• Apriori알고리즘 : 최소 지지도 갖는 연관규칙 찾는 방법(함수)

① 최소 지지도 설정 ② 개별 품목 중 최소 지지도 넘는 모든 품목 탐색 ③ 2에서 찾은 개별 품목만을 이용하여 최소 지가지 품목 집합 찾음 ④ 위의 두 절차에서 찾은 품목 집합을 결합하여 최소 지지도를 넘는 3가지 품목 집합 찾음 ⑤ 반복적으로 수행하여 최소 지지도가 넘는 빈발품목 찾음

라. 연관 분석 장점 : 탐색적인 기법(if-then ㅍ현되는 결과 이해 용이), 강력한 비목적성 분석 기법(목적변수 없음), 사용이 편리한 데이터 형태(거래 내용에 대한 데이터 변환 없이 이용 가능), 계산의 용이성

마. 연관 분석 단점 : 상당한 수의 계산 과정(품목수가 증가할 경우), 적절한 품목의 결정(너무 세분화된 품목으로 연관규칙 찾으려하면 의미 없는 결과 나올 수 있음), 품목의 비율차이(상대적으로 거래량이 적은 품목은 포함된 거래수가 적어 규칙 발견 시 제외 가능성↑)

바. 순차 패턴 : 시간에 따른 구매 정보, 구매 시점에 대한 정보 필요

2. 실습
>inspect(apriori(Adult) 실행 결과 lhs 항목 없는 경우 rhs 무의미한 결과
>plot(adult.rules.sorted, method="scatterplot") 연관규칙 시각화 패키지(arulesViz)

4장 비정형 데이터 마이닝

제1절 텍스트 마이닝 : 다양한 포맷(웹 콘텐츠, PDF, 마이크로소프트 오피스 파일, 오라클 오픈 오피스 파일, XML, 텍스트 파일 등)의 문서로부터 데이터를 획득해 이를 문서별 단어의 매트릭스로 만들어 추가 분석이나 데이터 마이닝 기법을 적용해 통찰, 의사결정 지원 방법
- 사용자별, 시간별 어떤 내용 언급하는지, 긍정적, 부정적 말을 하는지, 시간의 흐름에 따라 키워드 변화가 어떻게 되는지, 캠페인 진행 전후 고객의 키워드 변화, 프로모션 후 구전효과 유무, 어떤 집단의 고객들이 반응 체크, 경쟁사 브랜드 반응 모니터링

- 다양한 업무 영역 지원, 설비 고장 예측, 제품의 성능이나 불만사항 접수, 시간에 따른 주제의 흐름 변화, 특정 분야의 전문가 알아내는데 활용 가능

1. 텍스트 마이닝의 기능 : 문서 요약/분류/군집, 특성 추출(해당 언어에 대한 깊은 이해, 문화, 관스[q에 대한 이해 필요, 언어 및 국가별로 다른 접근 방식의 분석 수행]

2. 텍스트 마이닝의 기본 프로세스

가. 데이터 수집 : 트위터에서 자료 가져오는 방식 ①웹페이지에서 HTML 데이터로 가져와서 파싱(parsing) ②API 이용해 가져오는 방식(반드시 인증 거쳐 접근, 한번에 가져올 수 있는 데이터 크기 1,500건 제한, 쿼리를 이용해 일정시간 안에 지속적으로 많은 정보 가져오면 블로킹 등 제약)

>install.packages("twitteR") >library("twitteR") >keyword<-'bigdata'

>bigdata.tw<-searchTwitter(keyword, since='2014-01-01', n=1000, lang='en') #1,000개 메시지 지정했지만 일정시간 사용량에 따라 자동 제한되므로 가져오는 개수는 유동적

>tweet<-bigdata.tw[[1]]>bidat.df<-twListToDF(bigdata.tw) #list 형태의 bigdata.tw 데이터 프레임 형태로 변환

>bigdata.text<-laply(bigdata.tw, function(i) i$getText()) #plyr로 list→data frmae 형태 변환 가능

>bigdata.text<-bigdata.df$text#텍스트만 추출하여 저장

나. 데이터 전처리 및 가공

1) Corpus(데이터의 정제, 통합, 선택, 변환의 과정을 거친 구조화된 단계)생성 : Vcorpus(Volatile 메모리에서만 유지), Pcorpus(Permanent R 외부의 DB나 파일로 관리)

>txt<-system.file('texts', 'txt', package='tm') #읽어들일 문서의 경로

>ovid<-Corpus(DirSource(txt), readerControl=list(language='lat'))

>my.corpus<-Corpus(VectorSourxe(bigdata.text)) #bigdata.text text 부분만 추출한 벡터형태

>ovid[[1]] 개별문서 조회 [[ ]]로 인덱스 입력하여 조회

2) tm_map 함수 적용 : Corpus 형식 데이터들의 변형

tm_map(my.corpus, ~) →~ : removeNumbers, removePunctuation(문장부호, 구두점), removeWords, stemDocument, stripWhitespace(빈 공간), content_transformer('tolower'), PlainTextDocument

①빈공간 제거 :>my.corpus<-tm_map(my.corpus, stripWhitespace)

②특정 패턴을 가지는 단어 제거 또는 대체 : ex) @로 시작하는 리트윗 ID, http로 시작하는 URL 제거 >my.corpus<-tm_map(my.corpus, content_transformer(gsub), pattern='@\\S*', replacement='')>my.corpus<-tm_map(~), pattern='http\\S*', ~)

③문장 부호 및 구두점 제거 :>my.corpus<-tm_mat(my.corpus, removePunctuation)

④대문자를 소문자로 변경(한글이나 특수문자 오류 발생하므로 미리 처리한 후 진행) :>my.corpus<-tm_map(my.corpus, content_transformer(tolower))

⑤특정 단어 제거(조사) :>my.corpus<-tm_map(my.corpus, removeWords, stopwords('en')

다. 자연어 처리 : 형태소 분석(공통 어간 단어 묶음) stemDocument↔stemCompletion

1) Stemming

>my.corpus<-tm_map(my.corpus, stemDocument)

stemCompletion할 때 잘못하면 모든 값이 NA가 되어버리는 경우가 있어 completion 위해 간단하게 제작된 함수 실행

>stmeCompletion_mod<-function(x, dict) { PlainTextDocument(stripWhtespace (paste (stemCompletion(unlist(strsplit(as.character(x), "")), dictionary=+dict, type='first'), sep="", collapse= ""))) } #type='first'로 지정하여 등장하는 첫번째 어휘로 어간이 같은 모든 단어 사용하도록 설정

>my.corpus<-lapply(my.corpus, stemCompletion_mod, dict=dict.corpus) #모든 트위터 메시지에 stemCompletion_mod 적용

>my.corpus<-Corpus(VectorSource(my.corpus)) #반환된 list를 다시 Corpus로 변환

2) 한글 처리(KoNLP 패키지 사용 시 rJava 패키지, JRE 설치해야 함)

>extractNoun('명사 추출 연습을 해보고자 한다.') >sapply('명사 추출 연습을 해보고자 한다.')

라. TDM 구축(TermDocumentMatrix : 단어(행), 트위터 메시지(열) 매트릭스)

1) 모든 단어

>my.TDM<-TermDocumentMatrix(my.corpus) >inspect(my.TDM[55:60,1:10] #처음 10개 메시지의 55~60번째 단어의 분포 확인

Non-/sparse entries : 1/59 Sparsity :98%(=58/59)

2) 단어 사전(dictionary) : 분석에 사용하고자 하는 복수의 문자들의 집합

>myDict<-c('bigdata', 'data', 'analyst', 'cloud', 'company', 'privacy', 'analytics', 'business')

>my.TDM<-TermDocumentMatrix(my.corpus, control=list(dictionary=myDict))

마. 분석 및 시각화

1) Association

>findAssocs(my.TDM, 'warehouse', 0.5) #warehouse와 연관성 0.5 이상인 단어 표시

• my.TDM은 apriori함수를 이용해 연관분석을 하기 위한 단일 처리형(transaction)으로 변환

>transaction_m<-as(terms.m, "transactions")

>rules.all<-apriori(transaction_m, parameter=list(supp=0.01, conf=0.5))

• 지지도 20% 신뢰도 10% 설정하면 전체 데이터 중 특정 단어의 출현 빈도가 20% 이상이고 그 단어가 출현했을 때 다른 어떤 단어가 출현할 확률이 10%이상인 규칙을 도출한다는 의미

2) 워드 클라우드

>my.TDM.m<-as.matrix(my.TDM) #매트릭스 형태로 변환

>term.freq<-sort(rowSums(my.TDM.m), decreasing=T) #행을 기준으로 모든 열의 값을 합하

여 각 단어에 대한 빈도수 계산, 내림차순 정렬
>wordcloud(words=names(term.freq), freq=term.freq, min.freq=15, random.order=F, colors=brewer.pal(8, 'Dark2'))

3) 감성분석=오피니언 마이닝 :긍정적인 단어가 얼마나 많은지
영어 긍정/부정 단어 파일을 읽어와 저장(웹에서 가져오거나 c함수로 추가 가능)
>pos.word=scan("~/Desktop/positive-words.txt", what='character', comment.char=';')
>pos.word<-c(pos.word, 'upgrade')
>score.sentiment<-function(sentences, pos.words,neg.words, .progress='none') {
require(plyr) require(stringr)
scores<-laply(sentences, function(sentence, pos.words, neg.words) {
sentence<-gsub('[[:punct;]]', '', sentence) [[;cntrl;]], //d+' 문장부호, 특수기호, 숫자 빈칸 대체
sentence<-tolower(sentence)
word.list<-str_split(sentence, '₩₩s+') #문장의 분리 기준이 되는 패턴 정의(₩₩s+ : 띄어쓰기 혹은 띄어쓰기 이상의 빈칸) words<=unlist(word.list)
pos.matches<-match(words, pos.words) neg.matches<-match(words, neg.words)
pos.matches<-!is.na(pos.matches) neg.mathes<-!is.na(neg.matches)
score<-sum(pos.matches)-sum(neg.matches) return(score)},pos.words,neg.words, progress=.progress)
scores.df<-data.frame(score=scores, text=sentences) return(scores.df)}

제2절 사회연결망 분석(igrape 패키지)

1. 사회연결망 분석 정의

가. SNA 정의 : 개인과 집단들 간의 관계를 노드와 링크로서 모델링해 그것의 위상구조와 확산 및 진화 과정을 계량적으로 분석하는 방법론(사회연결망 용어 Barnes(1954)가 처음 사용, 기존에는 그룹 간, 그룹 안에 집중하였으나 Barnes는 독립 네트워크 사이 관계 집중)
1) 집합론적 방법 : 각 개체들 간의 관계를 관계 쌍으로 표현
2) 그래프 이론을 이용한 방법 : 객체는 점(노드)으로, 표현, 두 객체 간의 연결망은 선
3) 행렬을 이용한 방법 : 개체를 행, 열에 대칭적으로 배치하여 연결 있으면 1, 없으면 0
- 행과 열에 같은 개체가 배열되어있으면 원모드 매트릭스, 다른 개체가 배열되면 2원모드 매트릭스
- 국내 SNA에 가장 많이 활용되는 기법은 중심성(Centrality), 밀도(Density), 중심화(Centralization) 등이 있으며 중심성 측정 방법 대표적으로 4가지 있음

① 연결정도 중심성D(degree centrality) : 한 노드에 직접적으로 연결된 노드들의 합 ② 근접 중심성(closeness centrality) : 간접적으로 연결된 모든 노드 간의 거리를 합산, 한 노드로부터 다른 노드에 도달하기까지 필요한 최소단계의 합 ③ 매개 중심성(betweenness centrality) : 한 노드가 연결망 내의 다른 노드들 사이의 최다 경로 위에 위치하면 할수록 그 노드의 매개 중심성↑ ④ 위세 중심성(eigenvector centrality) : 연결된 노드의 중요성에 가중치를 두어 노드의 중심성 측정, 위세가 높은 사람들과 관계가 많을수록 자신의 위세↑

나. SNA 적용
- 노드는 고객, 에지는 고객과 고객 간의 관계(방향성 유무에 따라 방향/무방향 그래프 구분)
- 데이터 처리 속도 같은 기술적인 문제로 활용상 어려움 존재
- 분산 처리 기술인 하둡 MapReduce 활용하거나 하둡 기반의 그래프 프로세싱 프레임워크인 Giraph로 대용량 소셜 데이터를 R에서 처리가능한 수준으로 정제한 후 분석 수행 가능

다.SNA 단계 : 그래프 생성 단계, 목적에 따라 그래프 가공 분석하는 단계, 커뮤니티 탐지하고 각 객체, 노드의 롤을 정의해 어떠한 롤로 효율적으로 영향력을 줄 수 있는지 정의하는 단계, 이를 데이터화하여 다른 데이터 마이닝 기법 연계하는 단계로 구분

2. R을 이용한 SNA 활용 : 몇 개의 집단으로 구성되는지, 집단 간 특징, 영향력 있는 고객, 시간의 흐름과 고객 상태의 변화에 따라 다음 영향을 누가 받는지 등

가. 단어 간 연관성을 이용한 사회연결망 분석
>term.freq<-sor(rowSums(my.TDM.m), decreasing=T)
>my.Term<-my.TDM.m[rownames(my.TDM.m)%in%names(term.freq[term.freq>20]),] #빈도수 20 이상인 단어들로 이루어진 데이터만 추출하여 my.Term에 저장
>my.Term[my.Term>=1]<-1 #해당 단어가 사용된 횟수 1번 이상이면 1, 아니면 0 변환
>termMatrix<-my.Term%*%t(my.Term) #각 단어들 행, 열에 대칭으로 배치하여 한 단어가 다른 단어와 얼마만큼 함께 사용되었는지 파악
>g<-graph.adjacency(termMatrix, weight=T, mode='undirected') #그래프의 방향성은 단어의 연관성과 상관이 없으므로 undirected로 변환하여 실행
>g<-simplify(g) #loop나 multiple edge를 제거해서 단순화
>V(g)$label<-V(g)$name #V함수는 점, 노드에 대한 것, E함수는 에지에 대한 함수
>V(g)$degree<-degree(g) #소셜 네트워크에서 해당 노드, 점이 몇 개의 노드와 연결되어 있는지 나타내는 값인 연결정도(degree) 생성
>layout1<-layout.fruchterman.reingold(g) #의미있게 각 노드 배치해주는 함수
• 커뮤니티 : 관련성이 높은 집단(유력자>리더>브릿지 순으로 마케팅 실시)
• 유력자(influencer) : 자신의 커뮤니티와 다른 커뮤니티에 모두 연결이 많은 노드
• 리더 : 여러 노드와 연결된 것
• 브릿지 : 커뮤니티와 커뮤니티를 연결하는데 사용되는 노드
• passive : 커뮤니티 끝단에 있는 노드로 다양한 노드와 연결되지 않은 것

나. 트위터 검색을 통한 사용자 간 소셜 네트워크

노드 추가>g<-add.vertices(g,nrow(tw.names), name=as.character(tw.names$user), tweets=tw.names$tweets)

5과목 데이터 시각화

제1장 시각화 인사이트 프로세스

제1절 시각화 인사이트 프로세스의 의미

1. 인사이트란 무엇인가?
가. 인사이트의 사전적 정의 : cause and effect(인과관계), intuitively, noesis(초감각적), 정보, 본질, 이해
나. 데이터, 정보, 지식, 지혜, 관계
• 시각이해의 계층도 : 데이터(개별적인 요소 하나하나)-시각화, 정보(상관관계/인과관계 연관된 요소들)-디자인, 지식(조직화된 정보)-매핑, 지혜(개인의 경험/사고/감정 체계와 결합 적용된 지식, 개인화된 지식)-?
ex) 개별 강수량(데이터), 공간과 시간 관계 고려해 강수량 재구성하여 지역별 연간 최대 강수량(정보), A마을의 수해대책 매뉴얼(지식), 주민별 생활방식, 가치관 적용된 노하우(지혜)

2. 시각화와 인사이트 : 인사이트는 구체적인 것에서부터 시작
가. 삼찰 :관찰(외부 세계의 대상과 그 대상들 사이의 상호 작용을 통해 의미있는 관계 찾아내는 것), 성찰(자신의 내면 세계 살펴보는 것), 통찰(내부와 외부 요인들 간의 관계 탐색)
나. 통찰 과정과 시각화
1단계 탐색(시각화 형태 :**패턴 파악**)
- 특정한 방향 중심으로 탐색하면 의도한 관계를 파악할 수 있을지 모르지만 숨은 다른 관계는 놓칠 수 있음
- 탐색 과정에서 찾아낸 관계의 양상에 따라 분석할 관계의 범위와 우선 순위 결정
2단계 분석(**그래프 분석**)
- 분석의 방향성이나 검증 명제, 찾아야할 모델링(함수), 지표의 개요가 명확해야함
- 분석 단계에서는 정성적 기법 외에도 수치분석과 같은 정량적 기법 많이 사용
3단계 활용(검증, 보완, **인포그래픽**)
- 인사이트를 이용하는 대상에 따라 내부활용(찾아낸 특정한 원리나 모델을 서비스, 제품의 구동 원리에 직접 반영), 외부활용(도출한 통찰을 다른 사람에게 설명하거나 설득-스토리텔링 시각화 도구 : 인포그래픽)

3. 예시 : 땡처리 상품(숙박권을 모바일 기기에서 조회하고 구입하는 App, MO웹 서비스)
- 본 시각화 인사이트 프로세스의 목표 : 매출 증대를 위한 온라인 상품목록 최적화 등

제2절 탐색(1단계)

1. 사용 가능한 데이터 확인

가. 데이터 명세화 : 차원(값이 측정된 기준)과 측정값
- ex) 국가별 남성의 평균 수명 데이터(값 : 실수 형태의 나이, 차원 : 성별, 국가)
- 측정값을 분류할 수 있는 모든 것이 차원이 된다. 특정 시간대에 매초 기록한 측정값 같은 연속적인 데이터로 구성된 차원도 있다. 이런 연속 데이터 차원은 추후 탐색과 분석의 편의를 위해 구간 형태로 재구성되기도 한다.(시간, 일, 연 단위)
- 차원과 측정값의 구분은 분석 형태에 따라 달라질 수 있음(동일한 데이터 항목이라도 차원이 될 수 있고 측정값이 될 수 있음)
- 데이터의 형태 : 정수형(비연속 수치), 실수형(연속형 수치), 문자형, 날짜형
ex) 상품 코드(종류)에 따른 결제금액 : 차원-코드, 측정값-금액 / 결제금액을 구간화해 결제금액 크기별 상품 코드의 종류나 개수 분포 경우 : 차원-금액, 측정값-코드 개수 or 코드명

나. 데이터 구성 원리 1 : 이벤트 기록으로서 접근
- 원본데이터, 로그데이터는 특정 이벤트가 발생했을 때 생성된다. 이벤트는 반품과 같이물리적인 실제 사건일 수도, 강설량 10츠 넘기면 경보 울리기와 같이 인위적인 사건일 수도
- 온라인 서비스는 '순간 동시 접속자 수'(특정 시점의 서비스 이용자 수)나 '일일 액티브 이용자 수'(하루 기준으로 반복 접속을 제외한 모든 서비스 이용자 수)가 활성화 정도를 가늠하는 중요한 데이터이다. 로그데이터는 한 번 더 정제한 데이터(핵심 이벤트는 '접속'에 따라 로그인한 이용자의 ID, 시각, 기기의 종류 등)

다. 데이터 구성 원리 2 : 객체 지향 관점에서의 접근
- 만약 이벤트 로그가 부족하거나 없는 대신 살펴봐야할 대력적인 범위가 있다면 데이터의 구조 자체를 설계, 생성 → 객체지향 관점
- 객체지향론에서는 대상을 객체화하는데 모든 객체들은 행위와 고유속성값을 갖는다.
- 데이터의 구조 육하원칙 관점에서 ex) 시험A(대상), 학생(주체), 연도/학기/중간고사(시기), 학교, 시험장(장소), 과목(속성), 홍길동이 B중학교 교실에서 13년 1학기 중간고사로 치른 수학시험이 고유 객체(Object), 시험A가 오브젝트, 시험이라는 구조 자체는 클래스 해당
- 해당 시험 제도를 직접 만들어내고 운영하지 않는 상황에서 결과값만 보고 구조 전체 파악 불가능

- 스토리텔링 소화하기 위해서는 데이터 구분에서 상품에 대한 액션(method)인 전시, 조회, 예약 뿐만 아니라 상품 자체에 대한 정보도 알 수 있는 구조여야 함(상품이라는 클래스는 호텔 등급과 같은 구체적인 속성값을 가지며, 이 속상값들을 통해 상품이 결정되고, 각 상품코드는 하나의 완결된 오브젝트를 구분하는 대표값이다. 상품 클래스는 전시, 조회, 결제, 리뷰라는 행위(method)를 가지며, 각 행위는 다시 그에 따른 속성값들을 가진다.

2. 연결 고리의 확인 : 연결고리는 시각화 도구로 살펴보는 패턴에서 찾는 것이 아니라 데이터의 태생을 정리한 명세서에서 직접 확인, 데이터 확인 작업을 대폭 생략하고 주어진 데이터에서 바로 드러나는 관계만 살펴볼 수도 있지만, 이 경우 통찰의 질과 양 저하, 특히 살펴볼 데이터를 정의해야 하거나 서비스나 사업을 새로 설계하는 경우 반드시 데이터 구성 원리에 입각해 데이터 자체에 대해 고민한 다음 연결고리 확인해야한다.

가. 공통 요소 찾기 : 측정값을 기준으로 공통값을 가진 항목 탐색(데이터의 항목명이라는 기준 대신 해당 항목의 정의와 데이터형을 보고 찾아야 함)

나. 공통 요소로 변환하기 : 더 자세한 자료를 덜 자세하면서 묶인 자료로 변환하는 것 가능하나

반대로는 불가능
1) 시간 데이터의 변환
2) 공간 데이터의 변환 : 주소, 행정구역, 가장 구체적인 좌표값과 같은 공간 데이터가 있으며, 좌표계를 주소 및 행정구역으로 변환해주거나 그 반대로 변환해주는 것을 지오코딩
3) 일정한 규칙을 가진 분류형 데이터의 변환(vlookup, lookup 함수)
- 항목명이나 데이터형과 정의 자체가 동일한 것은 상품코드가 잇다. 상품코드를 통해 개별 상품에 대한 상품 스펙, 전시 현황, 조회 현황, 판매 현황, 리뷰 현황과 같은 모든 관점의 데이터로 연결된다. 구분된 모든 데이터 집합에는 공통적으로 날짜형, 시간형 데이터가 있는데 공통 요소 변환을 통해 날짜라는 공통의 연결고리(그룹화) 만들 수 있다. 이를 통해 특정 제품이 생성, 전시, 조회, 판매, 리뷰되는 전체 라이프사이클 패턴 분석 가능하다. 지오코딩 툴 통해 경위도 좌표계를 행정 지역 구분으로 묶어 공간 정보 연결고리 만들 수 있다.
- 모든 데이터는 데이터 구성 원리 의해 시간과 공간 관점의 연결고리 기본으로 갖고 있음

다. 탐색 범위의 설정 : 현재 데이터를 어떤 조합의 차원과 측정값으로 설정할지 고민
- 아직 명세화되어 있지 않은 데이터라면 명세화를 통해 사용 가능한 차원과 측정값을 찾고, 여러 개의 데이터 명세를 가지고 있다면 연결고리를 확인하여 명세서들을 포괄해 탐색할 수 있는 차원과 측정값의 조합 정리(관점에 따라 차원과 측정값 서로 바뀔 수 있음)
- 데이터를 구성하는 항목이 많을수록 탐색 범위를 설정할 수 밖에 없음(우선 순위 설정처럼) → 개별 데이터 안에서 먼저 탐색 후 데이터 간의 연결 고리를 이용해 전체 데이터 집합 안에서의 탐색 범위 설정, 처음에는 측정값에 하나의 차원만 연결해 탐색하고 점차 차원을 늘려가며 살펴본다. 차원, 측정값 맞바꾸어보기도 하며 목표에 관련됐을 법한 조합 탐색(상식적으로 의미나 연계성 없는 조합은 가급적 배제)

3. 관계의 탐색 : 인과관계는 상관관계 중에서도 명확하게 원인과 결과의 시간적 선후관계가 있는 관계이며 인과관계가 있는데 상관관계가 없을 수 있음(먼저 상관관계부터 탐색)

가. 이상값 처리
- 이상값 발생 경우 :① 데이터 측정 시 ② 데이터 기록, 관리 시 ③ 의미 있는 이유가 있어서 발생(1, 2번의 경우 오류 사항 제거하지만 3번의 경우 중점 탐색하여 인사이트 획득)
- 2번의 경우 문자형이나 다른 형태의 데이터가 있다면 대체하거나 제거하면 되지만 1, 3번의 경우 구분이 어렵다. 일단 구조적으로 불가능한 범위의 값이 기록됐다면 제거의 대상으로 간주한다. 보통 시각화 도구를 통해 전체 패턴을 살펴보고 처리(산포도)

나. 차원과 측정값 유형에 따른 관계 파악 시각화(1차원, 2차원, 3차원)
- 차원은 반드시 평면과 공간을 구성하는 축으로만 표현되는 것은 아니다.(면적도 연속값으로 된 차원 처리할 수 있는 도구, 3차원에서는 입체의 부피 또는 단면의 면적)
- 색상은 구분값으로 된 차원 처리하는 유용한 방법이다.(그라데이션 변화로 표현 가능)
- 시간 데이터에서의 관계 탐색은 시간에 따른 패턴의 변화 살펴보는 것이다. x축에 시간을 설정해 평면이나 공간상에 데이터를 뿌려 어떤 모양으로 전개되는지 왼쪽에서 오른쪽으로 훑어보는 방법과 동적인 시각화 도구인 모션차트 활용 가능
- 공간, 위치 데이터에서의 관계 탐색은 해당 위치를 표시하는 실제 지도 활용하면 효과적이며, 공간시각화 도구 파워맵(시간의 흐름에 따라 지도상의 데이터가 어떻게 변화하는지 모션차트 결합한 형태) 사용
- 비정형 데이터의 경우 문장 안의 단어들이 어떤 빈도로 분포하는지 시각화하는 도구 워들(Wordle) 활용하여 관계 탐색
- 워들을 만드려면 텍스트 데이터에서 형태소 단위 추출(자연어 처리 NLP 관점)하여 계산한 빈도에 따라 색상이나 크기 결정하여 겹치지 않게 배치해야 한다. 텍스트의 빈도 패턴을 탐색했다면 그 텍스트들의 의미, 품사, 긍정/부정 구분 등올 묶어 살펴본다.

다. 잘라보고 달리 보기(엑셀의 피벗/파워뷰 기능, OLAP-Business Intelligence 도구로 슬라이스 다이스 하며 분석, 리포팅하는 도구)
- 연령별 성별 평균 체중 데이터에 대해 20세 이상 40대 미만 남자들의 체중은 전체 패턴에서 어떻게 다른지 잘라보기(Slice)
- 여러 차원이 있는 경우, 차원들을 기준으로 잘라내 서로 다른 관점의 단면을 살펴보는 달리보기(Dice)

라. 내려다보고 올려다보기(피벗, 트리맵, 하이퍼볼릭 트리)
- 계층형 구조 없더라도 데이터 형태에 따라 묶는 작업을 통해 해당 구조 만들어낼 수 있다.
- 현재 바라보는 관점에서 하위 계층으로 기준을 세분화하는 내려다보기(Drill down, 미시적으로 보기)와 반대로 현재보다 상위 계층의 관점에서 보는 올려다보기(Reverse drill down, 거시적으로 보기)를 통해 계층형 구조 탐색
- 슬라이스는 어떤 차원을 기준으로 살펴보느냐에 대한 것이지만, 드릴다운은 계층구조를 형성하는 하나의 차원 안에서 세부적인 하위 차원으로 분할해보는 것(드릴다운은 슬라이스의 한 형태) ex) 시각 데이터는 데이터 변환을 통해 일/주/월/분기/연 단위의 상위 계층으로 올라가 볼 수 있으며 위치 데이터도 좌표계에서 구군단위, 시도단위, 국가/대륙 식으로 계층 묶어볼 수 있다.

마. 척도의 조정

- 인사이트 프로세스의 탐색 단계의 핵심은 개별값보다는 전체적인 패턴을 살펴보며 가능한 관계를 탐색하는 것(계열별로 최대값을 100으로 설정한 다음 동일 계열 내의 다른 값들을 이 비율에 맞춰 변환하여 동일한 공간에서 각각의 패턴 변화 비교) → 스파크라인 차트, 구글 스프레드시트(축에 절대값 표시)

제3절 분석(2단계)

1. 분석 대상의 구체화
가. 2차 탐색 : 어떤 패턴을 더 자세히 볼지 우선순위 결정
나. 분석 목표에 따른 분석 기법 :2차 탐색 과정을 거치며 시각화 도구로 인사이트 발견할 수도 있지만 통계적 분석 방법을 통해 패턴의 확률적 검증, 핵심 모델 도출 가능(차원이 너무 많거나 불연속 데이터의 수가 많다면 통계적 분석법 활용)

분석목표	통계적 분석 기법	분석목표	통계적 분석 기법
평균에 대한 검정과 추정	T검정	비율에 대한 검정과 추정	직접확률계산법, F분

			포법
분할표의 검정	카이제곱검정, 피셔의 직접확률 검정, 맥네마의 검정, 잔차 분석	변수들 간의 상관관계의 강도 추출	상관분석
변수들 간의 선형/비선형 인과관계의 형태와 강도 추출	회귀분석, 다중회귀분석, 로지스틱회귀분석, 판별분석	어떤 결과에 영향을 미치는 요인들 사이의 관계와 핵심 요인의 선별	요인분석, 주성분 분석
대상들을 여러 기준값들에 따라 분류하고 다차원 공간에 배치	군집 분석, 다차원척도법(MDS)	차원값들의 패턴이 비슷한 측정값과 그렇지 않은 측정값 분류	대응분석
시간의 흐름에 따라 변하는 데이터를 분석할 수 있는 모델 도출	시계열분석		

• 통계적 분석 기법의 결과물 : 모델 계수, 설명계수, 그래프, 걸러진 변수
• 최근에는 원본 모수 데이터를 구하기 쉬워 적용해야하는 통계적 분석 기법도 표본을 대상으로 하는 것이 아니라 모수를 대상으로 하는 경우가 많다.

2. 분석과 시각화 도구 : 통계적 분석 기법과 시각적 분석 기법은 서로 분리되지 않고 밀접하게 관련을 맺고 상보적으로 도움이 되는 관계

3. 지표 설정과 분석
• 지표 : 어떤 현상의 강도를 평가하는 기준이 되는 숫자(KPI 핵심성과지표) ex) 강수확률
가. 지표의 기본 구조 : 지표는 기존 값들을 어떤 함수식에 적용한 결과값
나. 지표 활용 시 주의점
• 지표는 원본 데이터에 추가되는 새로운 데이터지만 지표를 구성하는 다른 원본 데이터에서 비롯된 것이기 때문에 그들과 강력한 인과관계를 맺고 있는 특수한 성격의 데이터
• 지표의 단위 주의해야한다. 현실적으로 어떤 의미를 지니는지 의미
• 지표를 시각화 도구에 적용할 때 지표의 단위와 시각화 도구 표현 공간 상의 다른 데이터들과 함께 적절하게 표현될 수 있는지 체크해야하며 척도와 관련된 문제도 없는지 확인
• 다른 변수들이 지표와 어떤 관계에 있는지 검토해야한다.(지표를 만들어낸 변수와 지표가 통계적 모델을 만드는데 같이 들어간다면 모델의 설명력 과대 평가될 수 있으며 이런 경우 요인 분석을 해보면 이 지표가 지표를 만든 다른 요인과 설명력이 겹침을 알 수 있다.)
ex) 할인율, 할인가를 정상가로 나눈 값을 백분율화한 지표(지표를 구성하는 할인가, 정상가는 단위가 원이지만 지표 할인율의 단위는 없음)

제4절 활용(3단계)

1. 내부에서 적용 : 개인 또는 조직 단위로 실행(기존 문제 해결 방식이나 설명 모델의 수정, 새로운 문제 해결 방식 도입, 새롭게 발견한 가능성에 대한 구체적인 탐색과 발전)
• 통찰을 도출한 시점부터 시각화하여 동기 부여
• 통찰 활용하여 실행 시 자료가 실행과 관련된 현실적인 여건을 반영했는지 검토

2. 외부에 대한 설명, 설득과 시각화 도구
• 설명 대상과 그것을 도와주는 시각화 도구는 유기적인 연결고리를 가져야 한다.
• 설득의 경우 설명보다 강력한 상호작용이 필요하다.(더 강한 인과관계와 감성적 요소) → 사람의 마음을 움직이는 디자인(정보디자인, 인포그래픽)

3. 인사이트의 발전과 확장

가. **탑다운**(의미 있는 것을 몇가지 파악한 후 그것을 적용하는 과정에서 추가로 얻어낸 정보들을 토대로 검증하고자 하는 명제들을 명확하게 건증해나가는 방식) vs **보텀업**(새로운 대상에 대해 처음으로 무언가 살펴볼 때 아무것도 모르는 것을 전제로 하고 밑바닥에서부터 다양한 가능성 찾아보는 것)

나. 2차 잘라보기/달리보기, 내려다보기/올려다보기
• 시도해보지 않았던 차원들 간의 조합이나 특정 차원을 특정 값으로 고정해 살펴보며 인사이트를 고도화하고 확장

다. 실시간 vs 비실시간
• 보통은 일정 기간 누적된 기존의 데이터를 살펴보며 인사이트 도출하는 경우가 많다. 도출된 결과를 적용한 다음 효과를 검증하기 위해서는 적용 후 데이터를 다시 추출하여 적용 전긔 데이터와 비교해야한다. 비교 반복 주기에 따라 실시간으로 탐색하고 분석할 수 있는 환경을 구축하는 것이 나은지, 비실시간이라도 주기적으로 새로운 데이터 축적해 살펴볼 수 있는 환경을 구축하는 것이 나은지 결정하고 진행하는 작업 필요하다.
• 변화의 경향을 주기적으로 살펴보는 것이라면 굳이 실시간 처리 환경 아니어도 된다.
• 실시간으로 처리해야 하는 것은 긴급한 위기상황을 처리하기 위한 모니터링 및 경보 시스템의 경우이며, 이럴 때는 지표운영에 중점을 두어야 한다.

라. 지표 운영
• 지표 : 관계를 하나의 수치로 표현해 분석에 활용할 수 있는 형태로 전환한 것(몇가지 지표만 집중해봐도 다양한 관계들을 통해 나타나는 전체적인 흐름 파악 가능)
• 인사이트 프로세스에서 추출한 지표 중심으로 운영할 경우 문제점 : 환산된 값을 중심으로 보다보니 정작 어떤 변화요인이 발생해 지표 흐름에 영향을 미쳤는지 파악 어려움(조직의 상황에 따라 지표 운영과 인사이트 프로세스 실행이 분리 운영되는 경우 운영자 관점에서는 지표의 변동이 어떤 의미를 갖는지 모를 수도 있다.)

마. 추가 데이터에 대한 필요성 : 통찰, 추출 담당자와 데이터 운영 담당자 다른 사람인 경우 데이터 추가에 대한 갈등 발생할 수 있다.

바. 시각화의 오류 : 척도에 대해 이해하지 못하면 의미 있는 패턴 놓칠 수 있다. 값을 어떻게 묶느냐에 따라 같은 시각화 도구에서도 완전히 다른 모양의 패턴 나올 수 있다.
ex) 눈의 착각, 3차원 공간에서 무언가 표시할 경우 원근감 때문에 잘못 해석할 수 있으며, 크기나 색상이 다양하게 나타내는 요소들이 중첩적으로 들어가는 경우 잘못 해석 가능

사. 사람의 문제 : 통찰은 개인차가 발생할 수 있는 영역

ex) 한 번 구매한 이용자의 두번째 구매 이후부터의 구매 패턴과 최초 구매 이용자의 구매패턴에서 차이가 난다고 생각 → 두 번 이상 구매한 고객과 최초 구매 고객 집단 분리 → 기존 구조에서 회원제 전제로 한 분석 서비스가 아니어서 상품 결제 시 입력하는 이름, 전화번호, 이메일 주소 등으로 회원 구분하며 고객별 구매 횟수 기록 → 구매 횟수를 하나의 차원으로 설정하고 다른 차원에는 호텔 등급, 지역, 시간 등의 조합을 넣고 측정값에는 호텔 이름을 두어 호텔 개수를 비교하거나, 건별 결제 금액을 두어 평균 결제 금액의 분포를 비교하는 작업 → 비주얼 인사이트 프로세스를 거쳐 로열티에 따른 구매 패턴 인사이트 발견 → 해당 인사이트 매출 확대를 위해 활용하기 위해서는 회원가입 베이스로 서비스 구조 및 로그 데이터 구조 업그레이드 해야한다. → 회원에 따라 상품 배치 개선, 이벤트 진행, 기타 차별적인 요소에 대해 전략적인 의사결정

제2장 시각화 디자인

제1절 시각화의 정의

1. 데이터 시각화의 중요성
• 빅데이터의 가장 큰 특징은 텍스트와 이미지가 비정형성을 가지고 있으며, 규모 뿐 아니라 빠르게 전파되기 때문에 중요한 패턴을 찾기 쉽지 않다.
• 유용한 정보의 증가만큼 불필요한 정보도 급증하여 방대한 데이터 속에서 의미를 찾아내고 분석하는 일이 중요해졌다.(할 배리언이 앞으로 10년간 중요한 능력 5가지 : 데이터 얻는 능력, 처리 능력, 가치 추출 능력, 시각화 능력, 전달 능력)
• 데이터 시각화의 목적 : 데이터 분석 및 의사소통
• 데이터 시[각화를 통한 효과 : 자료로부터 정보 습득하는 시간의 절감, 즉각적인 상황 판단사름들의 흥미 유발, 정보의 빠른 확산, 기억 용이

2. 시각 이해와 시각화 p677
가. 데이터 : 결론을 내리는데 근거가 되는 사실이나 참고 자료, 연구나 조사, 발견, 수집의 결과인 일종의 기초 자료, 정보로서의 가치가 부족하며 데이터를 만들어낸 생산자에게는 유용할 수 있으나 사용자에게 의미 전달하기에는 부적절, 데이터 그 자체가 우리가 디자인하려는 대상은 아니다.
나. 정보 : 그 자체만으로도 유의미, 생산자와 사용자의 관점에 따라 다르게 전달될 수 있으며, 나름대로의 형태와 형식을 갖추고 있다. 서로 다른 데이터 간의 관계와 일정한 패턴을 가시화시켜 정보를 보는 사람에게 데이터가 내포하는 의미 전달한다. 데이터와 정보를 더 명확하게 구분하려면 정보 생산과 활용 과정에서 전체적인 맥락을 고려해야 한다. 정보는 생산자와 소비자의 두 영역에 모두 포함되면서도 자기조직화 하지 않은 일반적인 의미만 내재하고 있다.
다. 지식 : 지식은 자기조직화를 통해 국소 맥락(Local Context) 영역에 해당, 인간이 생활을 영위하면서 인위적으로 습득하는 곱도의 논리적 상식이자, 정보의 상위 개념이며 모든 경험의 산물, 다양한 상황에서도 적용할 수 있게 일반화한 것, 스토리텔링 개념 중요

라. 지혜 : 정보가 특정 영역에서의 경험에 의해 촉진돼 자기 맥락을 갖게 될 때 지식이 되며, 이런 지식은 자기 내면화가 되어 개인적 맥락(Personal Context) 안에 포함될 때 지혜가 된다. 정보와 지식의 개인화에 의해 생성되며, 인위적으로 전달하거나 공유할 수 있는 것이 아니다.(=메타지식)
• 공급자는 데이터와 정보의 단계에 속하고, 수용자는 정보와 지식의 단계에 속한다. 또한 정보는 글로벌 콘텍스트에 있지만, 지식은 로컬 콘텍스트에 있다.p679

3. 시각화 분류와 구분
가. 데이터 시각화 : 도식적 형태 안에 추상적으로 표현된 속성이나 변수를 가진 단위를 포함한 정보를 말하며 주요 목적은 그래픽의 의미를 이용해 명확하고 효괒거으로 정보를 커뮤니케이션하기 위함이다. 데이터들의 연결과 그룹핑을 표현하는데 중점을 두며, 통계적 그래픽, 주제 지도학의 관점이 있다. 데이터 시각화의 범위로는 마인드맵, 뉴스 표현, 데이터 표현, 관계들의 표현, 웹사이트들의 표현, 기사와 리소스들, 툴과 서비스 등이 있다.

나. 정보 시각화 : 정보 시각화는 보통 대규모 비수량 정보를 시각적으로 표현하는 것이며 정보 시각화 분야는 데이터 시각화 분야보다 한 단계 더 정보 형태로 가공 과정을 거치며 분기도, 수지도, 히트맵 등의 다양한 그래프를 통해 표현된다.

다. 정보 디자인 : 데이터 시각하ㄴ, 정보 시각화도 정보 디자인의 범위에 속하며 정보, 데이터 또는 지식 등 그래프 정보 표현 방법이 많이 적용되는 인포그래픽 역시 정보 디자인의 한 종류임(정보 그래픽 중 가장 의미있는 것은 나폴레옹 행군 다이어그램, 나이팅게일의 폴라그래프 등)

라. 인포그래픽 : 정보형 메시지(지하철 노선도), 설득형 메시지(가레스 홀트의 사회체제 묘사(Picturing social order) - 사회 계층을 암시하는 옷을 가로로 재단하고 상하 계층별로 다시 연결해 한 벌의 셔츠로 표현한 이 그래픽은 사회 계층별 분포 데이터를 극단적으로 함축하여 시각화)로 나뉘며, 인포그래픽은 원데이터를 취급하지 않는다. 대신 실용적으로 전달하기 위해 다양한 차트, 다이어그램, 일러스트레이션이 적극 사용된다.

4. 빅데이터 시각화 영역 : 데이터 시각화는 같은 범주 안의 많은 데이터에 의미 부여해 효율적으로 전달하기 위함이며, 정보 시각화는 큰 범주에 해당하는 정보를 시각화하는 것이다. 정보 디자인은 인포그래픽을 포함해 시각 스토리텔링 형식의 설득형 메시지를 전달하는 것에 초점

정보형 메시지←정보 디자인←→→→→설득형 메시지
데이터 시각화 / 정보 시각화 / 인포 시각화

빅데이터 시각화는 데이터를 기반으로 객관적 표현에 더 초점을 맞추는 경우가 많으므로 데이터를 직접적으로 전달하는 기능성에 초점을 맞춘 정보형 메시지를 전달하기 위한 데이터 시각화 작업을 하는 경향이 강하다. 정보 디자인에서는 원인과 결과의 인과관계를 왜곡없이 전달하는데에 초점을 두고 있다. 상대적으로 인포그래픽은 양적 정보 디자인(다차원적인 표현 추구)보다는 다양한 정보 종합해 정보 디자인 의도에 따라 그래픽으로 전달하려는 경향이 강하다.

제2절 시각화 프로세스

1. 정보 디자인 프로세스
1단계 데이터 수집 : 복잡한 데이터로부터 시작, 리소스들을 정보 제공자로부터 받아 데이터 수집 과정에서 스토리를 시작할 수 있는 단서들을 찾아낸다. 이 단서들로부터 새로운 리소스에 대한 추가 리서치 진행, 로우 데이터를 직접 수집하기는 어려우며 빅데이터의 풀을 탐험하는 과정
2단계 모든것을 읽기 : 리서치 에코시스템 안에서 정보의 작은 조각들을 큰 그림으로 짜 맞추기
3단계 내러티브 찾기 : 인포그래픽은 복잡한 데이터세트를 단순 명료화하고 프로세스를 설명하고 트렌드를 창조하고 논란의 어떤 부분을 보조하는 특별한 의도와 함께 시작된다.(스토리 찾기)
4단계 문제의 정의 : 현실성 검토해야 하며, 더 정확한 내러티브와 표현을 위한 탐험을 해야 한다. 더 주관적 관점에서 디테일을 만들어가야 한다.
5단계 계층구조 만들기 : 중요한 것은 주인공으로 만들고 나머지는 보조적인 요소로 배열
6단계 와이어프레임 그리기 : 시각화의 최종 결과물에 가까운 구조를 만들 수 있는 정보의 와이어프레임 제작
7단계 포맷 선택하기 : 전통적인 차트와 그래프(바, 선, 파이 등) 이용 및 프로세스 설명하기 위한 다이어그램, 흐름도, 인터랙션
8단계 시각 접근 방법 결정하기 : ① 데이비드 맥캔들레스나 니콜라스 펠톤의 초기 데이터의 아름다움 만들어내는 방법, 차트나 그래프 형태 ② 피터 온토트나 스코트 스토웰처럼 일러스트레이션이나 메타포 이용 방법, 피터는 사진을 이용한 시각화 이용 ③ 하이퍼 액트 하이브리드(①+②)
9단계 정제와 테스트 : 데이터 본 적 없는 사람에게 이해되는지 반복 테스트
10단계 세상에 선보이기 : 완성된 인포그래픽은 온라인을 통해 공유

2. 빅데이터 시각화 프로세스
- 정보 디자인 교과서 : ① 조직화한 데이터 ② 시각적 매핑 ③시각적 형태 ④ 전달 방식
- 마티아스 샤피로 : ① 질문 만들어내기 ② 데이터 수집 ③ 시각적 표현 적용
- 벤 프라이 : ① 정보 획득 ② 분해(prase) ③ 선별(filer) ④ 마이닝(mine) ⑤ 표현 ⑥ 정제(refine) ⑦ 상호작용 (표현 및 정제 단계 에서는 그래픽 능력 요구한다. 직접적인 시각 표현 단계에 해당된다.)
- 일반적인 빅데이터 시각화 과정
• 1단계 정보 구조화 : 데이터를 수집하고 정제하는 과정으로 데이터세트를 만들기 위한 분석 도구 필요(수집 및 탐색하기, 분류, 배열, 재배열 4단계로 나누기)
• 2단계 정보 시각화 : 주로 분석 도구에서 제공하는 그래프나 분석 도구의 특성에 따른 시각화(시각화 툴 이용 - 전문 시각화 툴 중 프로세싱은 인터랙티브 버전을 구현하는데 적합하지만 데이터 마이닝 부분은 포함하고 있지 않으므로 다른 통계 프로그램 함께 사용해야한다.)
• 3단계 정보 시각 표현 : 시각화의 의도를 강화해 전달하기 위해 분석 도구에서 만든 결과물에 별도 그래픽 요소를 추가해 최종적으로 시각적인 완성 단계

제3절 시각화 방법 : 정보의 구조화를 위해서는 정보를 분류, 조직화, 재배열 / 정보 시각화를 위해서는 각종 툴에서 일반적으로 제공하는 그래프 스타일의 원리와 쓰임새 이해 / 최종 완성을 위해서는 정보 시각표현에 해당하는 기초 조형과 그리드, 타이포그래피, 색상등의 원리 이해

1. 정보 구조화 : 원 데이터의 구문을 분석, 정리하고집단으로 묶거나 변환해 패턴을 식별하거나 특정 정보를 추출, 데이터 멍잉(data munging)
가. 데이터 수집 및 탐색 : 유의미한 데이터 선정하고 무의미한 데이터 걸러내는 데이터 에디팅
나. 데이터 분류
- 구분 텍스트 : 데이터의 줄바꿈으로 행을, 구분자로 열을 구분하는 텍스트 데이터(CSV파일의 구분자는 쉼표, TSV라면 탭으로 구분)
- JSON : 자바스크립트 객체 형식(JavaScript Object Notation), 배열과 복합 객체들 표현
- XMLS : 확장마크업 언어(eXtensible Markup Language), 문서를 사람과 기계 모두 읽을 수 있는 형식으로 부호화하는 규칙의 집합 정의(구조적 데이터 설명)
다. 배열 : 리처드 솔 워먼의 저서 “정보 욕구”에서 래치(LATCH) 방법 제시가 정보 정리, 조직화하는 기준)
- 위치(Location) : 지리적인 것만ㅇ 아니라 공간적으로 구분하는 것 모두 포함
- 알파벳(Alphabet) : 가나다순 정렬
- 시간(Time) : 일정 기간에 일어난 사건을 조직화하는 최적 원리는 시간으로 정보 배열하는 것
- 카테고리(Category) : 종류, 분류 의미하는 카테고리는 정보의 속성에 따라 분류할때 적합
- 위계/가중치(Hierarchy) : 정보의 변화에 따라 데이터의 값이나 중요도의 순서로 정보를 조직화, 카테고리에 의한 조직화와 달리 가중치는 단위나 수치로 표현 가능
라. 관계맺기(재배열) : 데이터 재배열은 데이터에 의미를 부여하는 가장 기본적인 과정으로 정보의 시각화와 밀접한 관계가 있다. 수용자가 인식하기 쉽게 패턴을 만드는 일

2. 정보 시각화 : 정보 시각화 방법은 분석과 함께 제공되는 시각화 도구(Tool)에 의해 결정되는 경향이 강하다. 효과적인 분석 디자인은 생각하는 작업을 보는 작업으로 변화시켜야 한다. 사고의 목적이 인과관계를 이해하는 것이라면 디자인의 목적은 인과관계를 보여주는 것이 된다.(에드워드 터프티)
가. 시간 시각화 : 시계열 데이터의 트렌드, 경향성 - (누적)막대그래프, 점그래프, 연결된 점, 선 그래프(연속형 데이터의 경우)
나. 분포 시각화 : 데이터의 특성에 맞게 전체의 관점에서 부분 간의 관계를 보여줘야 한다. - 원 그래프(데이터 분석에는 거의 사용X, 최대한 구성요소를 제한하고 내용을 설명하기 위한 텍스트와 %를 포함하는 것이 좋다. 부분간의 관계를 보여주며, 면적으로 값을 보여주고 수치를 각도로 표시한다.), 도넛차트(중심의 구멍때문에 조각에 해당하는 수치는 조각의 면적이 아닌 길이로 표시), 트리맵(위계 구조가 있는 데이터나 트리 구조의 데이터 표시), 누적 연속그래프(ex 네임 보이저, 어느 시기에 어떤 아기 이름이 얼마나 선택됐는지)
다. 관계 시각화 : 산점도, 버블차트(세번째 변수의 데이터는 버블 면적으로 표현), 히스토그램
라. 비교 시각화 : 히트맵(한 칸의 색상으로 데이터 값 표현, ex 마켓프로핏에서 제공하는 감정 히트맵-향후 주식시장에 대한 전망을 사회적 주식 지수로 보여준다. 트위테어서 많이 이야기되는 주식이 블록에 나타나며 해당 주식에 대해 어떤 감정으로 이야기하는지가 색상으로 나타난다.), 체르노프 페이스(데이터를 사람의 얼굴 이미지로 표현, 얼굴의 가로 너비, 세로 높이, 눈, 코, 입, 귀 등 각 부위를 변수로 대체, 엄밀한 의미의 데이터 그래픽에는 포함X), 스타차트(거미줄 차트, 방사형 차트), 평행 좌표계(여러 축을 평행으로 배치해 한 축에서 윗부분은 변수 값 범위의 최대값, 아래 변수 값 범위의 최소값 나타낸다. 대상이 많은 데이터에서 집단적인 경향성을 쉽게 알아볼 수 있다.), 다차원척도법(대상 간의 유사성/선호도 측도에 의거해 대상을 다차원 공간 속에 배치시키는 방법, 유사성이 작은 대상끼리는 멀리 유사한 대상끼리는 가까이 위치)
마. 공간 시각화 : 위도와 경도의 위치 값을 이용해 지도 위에 정확한 매핑 포인트를 표시해야한

다. 구글 차트의 지오차트는 이 값을 모르고 지명만 알아도 시각화 작업 가능하다.

- 에드워드 터프티는 데이터링크가 아닌 것과 중복되는 데이터링크를 제거해 데이터링크 비율을 올리는 것이 데이터를 그래픽 디자인으로 올바르게 표현하는 방법이라고 주장(①배경 지우기 ②범례 지우기 ③테두리 지우기 ④색깔 지우기 ⑤특수효과 지우기 ⑥굵은 글씨 지우기 ⑦라벨 흐리게 처리 ⑧보조선을 흐리게 또는 지우기 ⑨라벨(레이블) 직접 표시

3. 정보 시각표현
가. 정보표현을 위한 그래픽 요소 : 위치, 크기, 모양, 색, 명도, 기울기, 질감
나. 시각화를 위한 그래픽 디자인 기본 원리
- 타이포그래피 : 서체(돌기 있는 세리프 서체와 돌기 없는 산세리프 서체로 구분, 세리프 서체는 가독성이 높아 본문용, 산세리프 서체는 주목성이 높아 제목용), 무게(두께), 크기, 스타일, 색채, 간격(글자사이, 낱말사이, 글줄사이)
- 색상 : 보색을 추출하거나 유사색을 과학적으로 추출하기 위해 컬러스킴을 사용한다. 구분, 순서(ussk, 그라데이션 활용), 비율, 색채 사용과 인지 표현
- 그리드 : 그리드 이용해 블록 레이아웃을 잡고 그 위에 요소를 효율적으로 올려놓아 전체적인 조화 추구 중요(하나의 화면을 읽는 방식-상단 왼쪽에서 오른쪽 하단으로 / 정보의 역피라미드-가장 중요한 정보 맨 위 / 망 그리드-수평,수직선의 연속이 개체를 배치하는 지침이 되어 일관성 있고 정확하게 개체 배치 가능 / 3등분 법칙-3*3그리드 선이 교차하는 곳을 적극적 핫스팟으로)
- 아이소타이프 : 국제적인 그림 언어 체계로 단순히 우리 눈에 익숙한 픽토그램을 뜻하는 것은 아니다. 통계 그래픽에 뿌리를 두었으므로 하나의 기호가 일정 수량을 대표한다는 점이 중요

다. 인터랙션 : 사용자가 스스로 정보를 필터링하고 탐색하는 과정에서 인사이트 확보 기회를 제공한다. 대부분 인터랙션 위에 구현된 정보 디자인은 비선형적 구조이다. 정보 전달 과정에서 시간 제약이 없으므로 사용자의 정보 이용이 능동적이며 비교적 자세하게 탐색하므로 정보의 전달 효과 높다. 정보 메시지도 다양하게 조직화할 수 있으며 디지털 미디어의 상호작용 특성으로 정보의 폭과 깊이를 사용자가 통제할 수 있다. 디지털 환경에서 정보 전달을 위한 인터랙션은 사용자의 행동이나 조작에 따른 반응, 감각의 확장, 정보 시각화의 변화 등으로 정보와 정보 사용자 간의 관계를 확장하고 심화하는 것, 사용자 참여를 유도해 적극적으로 정보에 접근하게 하며 흥미 유발해 정보에 대한 관여도 높이는 역할
- 강조하고 디테일을 보여주는 방식 : 마우스 움직임에 반응하며 강조
- 사용자가 콘텐츠를 선택하는 방식 : ex 가족구성원을 더하거나 빼며 기본 가구타입 선택
- 여러가지 방법으로 데이터 보여주기 : ex 가디언 트위터에서 소문이 어떻게 퍼지나 시각화=버블차트)다른 트윗들, 이 트윗과 관련된 정보)+선그래프(소문과 관계된 시간당 트윗들과 중요한 이벤트 강조)
- 사용자 지정으로 시각 맵핑 변화(멀티 조정 시각화) : 동일한 시간대에 데이터의 다양한 관점 보여준다. 사용자가 시각화 유형을 지정할 수 있도록 시각 데이터 재매핑을 지원하며 시각화 크기 극대화 한다.
- 사용자의 관점과 의견이 반영된 형태 : ex OECD 더 나은 삶 지수 시각화-인터랙티브 버전으로 거주, 삶의 만족도, 교육 등 주제의 중요도를 사용자가 설정하여 각 주제 측정은 사용자가 생각하는 주관적인 지표에 의해 결정

라. 시각 정보 디자인 7원칙
- 시각적 비교 강화 / 인과관계 제시 / 다중변수 표시 / 텍스트, 그래픽, 데이터를 한 화면에 조화롭게 배치 / 콘텐츠의 질과 연관성, 진실성 분명히 하라 / 공간순 나열 / 정량적 자료의 정량성 제거하지 말 것

제4절 빅데이터와 시각화 디자인

1. 빅데이터와 시각화 이슈 : 조선일보의 시각화 결과물은 기사의 주장을 시각적으로 보조, 의도된 메시지 전달하는 성격이나 뉴욕타임즈의 시각화 결과물은 독자 스스로에게 데이터 탐험하게 해 패턴을 찾고 결론을 얻도록 하는 객관성 특징, 기업에서 빅데이터 시각화를 통해 제공 할 수 있는 것은 내부적으로는 방대한 정보를 하나의 인사이트로 도출할 수 있는 시각적 분석 도구로 활용하는 것, 외부적으로는 빅데이터를 이용해 도출한 정보를 고객에게 제공하는 것
2. 빅데이터와 시각화 디자인 사례 : 2D이미지(인쇄물, 온라인 이미지), 모션 영상(모션 인포그래픽, 데이터 시각화 동영상), 인터랙티브(인터랙티브 웹/앱)

제3장 시각화 구현

제1절 시각화 구현 개요 : 시각화 방법에는 시각화 플랫폼 이용하는 방법, 시각화 전문 라이브러리를 통해 직접 개발하는 방법, 데이터 표현을 위한 디자인 강화해 인포그래픽으로 표현하는 방법 등이 있다.

제2절 분석 도구를 이용한 시각화 구현 : R

제3절 라이브러리 기반의 시각화 구현 : D3.js

2과목 데이터 처리 기술 이해

제1장 데이터 처리 프로세스

제1절 ETL(Extraction, Transformation and Load)

1. ETL 개요 : 데이터 이동과 변환 절차와 관련된 업계 표준 용어로 데이터 웨어하우스(DW), 운영 데이터 스토어(Operational Data Store), 데이터 마트(DM)에 대한 데이터 적재 작업의 핵심 구성요소
- ETL 기능
• 추출(Extraction) : 하나 또는 그 이상의 데이터 원천들로부터 데이터 획득
• 변형(Transformation) : 데이터 클렌징, 형식 변환, 표준화, 통합 또는 다수 애플리케이션에 내장된 비즈니스 툴 적용 등
• 적재(Loading) : 위 변형 단계 처리가 완료된 데이터를 특정 목표 시스템에 적재
- ETL 구분 : 일괄(Batch) ETL, 실시간(Real Time) ETL
• Step 0 interface : 다양한 이기종 DBMS 및 스프레드시트 등 데이터 원천으로부터 데이터 획득하기 위한 인터페이스 메커니즘 구현
• Step 1 Staging ETL : 수립된 일정에 따라 데이터 원천으로부터 트랜잭션 데이터 획득 작업 수행 후 획득된 데이터를 스테이징 테이블에 저장
• Step 2 Profiling ETL : 스테이징 테이블에서 데이터 특성을 식별하고 품질 측정
• Step 3 Cleansing ETL : 다양한 규칙들을 활용해 프로파일링된 데이터 보정 작업
• Step 4 Integration ETL : (이름, 값, 구조) 데이터 충돌을 해소하고, 클렌징된 데이터 통합
• Step 5 Demoralizing ETL : 운영 보고서 생성, 데이터 웨어하우스 또는 데이터 마트 데이터 적재를 위해 데이터 비정규화 수행

2. ODS(Operational Data Store) 구성 : 데이터에 추가 작업을 위해 다양한 데이터 원천들로부터의 데이터를 추출, 통합한 데이터 베이스, 향후 타 정보 시스템이나 데이터 웨어하우스로 이관, ODS는 일반적으로 실시간(Real Time) 또는 실시간 근접(Near Real Time) 트랜잭션 또는 가격 등 원자성(개별성)을 지닌 하위 수준 데이터들을 저장하기 위해 설계
가. **인터페이스 단계** : 다양한 데이터 원천으로부터 데이터 획득하는 단계, 데이터를 획득하기 위한 프로토콜로는 OLEDB(Object Linking and Embedding Database), ODBC(Object Data Base Connectivity), FTP(File Transfer Protocol) 필요하며 데이터 웨어하우스에 대한 실시간/근접 실시간 OLAP(Online Analytical Processing) 질의를 지원하기 위해 실시간 데이터 복제 인터페이스 기술들이 함께 활용된다.
나. **데이터 스테이징 단계** : 작업 일정이 통제되는 프로세스들에 의해 데이터 원천들로부터 트랜잭션 데이터들이 추출되어 하나 또는 그 이상의 스테이징 테이블들에 저장된다. 이 테이블들은 정규화가 배제되며, 테이블 스키마는 데이터 원천의 구조에 의존적이다.
다. **데이터 프로파일링 단계** : 범위, 도메인, 유일성 확보 등을 기준으로 데이터 품질 점검
• 스테이징 테이블 내 데이터에 대한 데이터 프로파일링 수행→데이터 프로파일링 결과 통계 처리(Load Statistics→Profiling Statistics)→데이터 품질 보고서 생성 및 공유
라. **데이터 클렌징 단계** : 프로파일링 단계에서 식별된 오류 데이터들 수정
• 클렌징 스토어드 프로시저 실행(예비 작업)→클렌징 ETL 도구 실행
마. **데이터 인티그레이션 단계** : 수정 완료한 데이터 ODS 내의 단일 통합 테이블에 적재
• (데이터 충돌 판단 요건)→통합 스토어드 프로시저 실행(예비 작업)→통합 ETL 도구 실행
바. **익스포트 단계** : 익스포트 규칙과 보안 규칙 반영한 익스포트 ETL 기능 수행, 다양한 전용 DBMS 클라이언트 또는 데이터 마트, 데이터 웨어하우스에 적재

3. 데이터 웨어하우스
- 특징
• 주제 중심(Subfect Oriented) : 실 업무 상황의 특정 이벤트나 업무 항목을 기준으로 구조화
• 영속성(Non Volatile) : 최초 저장 이후 읽기 전용(Read Only) 속성을 가지며 삭제되지 않음
• 통합성(Integrated)
• 시계열성(Time Variant) : 시간순에 의한 이력 데이터 보유

가. 스타 스키마
• 조인 스키마라고도 하며 데이터 웨어하우스 스키마 중 가장 단순하다. 단일 사실 테이블(Fact Table)을 중심으로 다수의 차원 테이블(Dimensional Table)들로 구성, 다차원 데이터베이스 기능 구현 가능
• 이해하기 쉬우며 쿼리 작성이 용이하고 조인 테이블 개수가 적은 것이 장점이며, 스타 스키마의 사실 테이블은 보통 제3정규형으로 모델링하며 차원 테이블은 비정규화된 제2정규형으로 모델링하는 것이 일반적(차원 테이블을 정규화하는 것을 스노우 플래이크 스키마라고 한다.)
• 스타 스키마는 차원 테이블들의 비정규화에 따른 데이터 중복으로 해당 테이블에의 데이터 적재 시 상대적으로 많은 시간 소요되는 단점 존재

나. 스노우 플래이크 스키마 : 스타 스키마의 차원 테이블을 제3정규형으로 정규화한 형태로 데이터의 중복이 제거되어 데이터 적재 시 시간 단축 but 스키마 구조 복잡성 증가로 조인 테이블 개수 증가와 쿼리 작성 난이도 상승

제2절 CDC(change Data Capture)

1. CDC(Change Data Capture) 개요 : 데이터베이스 내 데이터에 대한 변경을 식별해 필요한 후속 처리(데이터 전송, 공유 등)를 자동화하는 기술, 실시간 또는 근접 실시간 데이터 통합을 기반으로 하는 데이터 웨어하우스, 저장소 구축에 활용된다. CDC 구현 시 데이터 원천에서 변경을 식별하고 대상 시스템(Target)에 변경 데이터를 적재해주는 '푸시'방식과 데이터 원천을 정기적으로 살펴보아 필요 시 데이터를 다운로드하는 '풀'방식 구분

가. Time Stamp on Rows : 변경이 반드시 인지되어야 하는 테이블 내 **마지막 변경 시점을 기록하는 타임스탬프 칼럼**을 두고 마지막 변경 타임스탬프 값보다 더 최근의 타임스탬프 값을 갖는 레코드를 변경된 것으로 식별한다.

나. Version Numbers on Rows : 변경이 반드시 인지되어야 하는 테이블 내 해당 **레코드의 버전을** 기록하는 칼럼을 두고 기 식별된 레코드 버전보다 더 높은 버전을 보유한 레코드를 변경된 것

으로 식별한다. 레코드들의 최신 버전을 기록, 관리하는 '참조 테이블'을 함께 운용하는 것이 일반적이다.

다. Status on Rows : 보완 용도로 활용, **변경 여부를 True/False 값으로 저장하는 칼럼**의 상태 값을 기반으로 변경 여부 판단

라. Time/Version/Status on Rows : 타임 스탬프, 버전 넘버, 상태 값 세가지 모두 활용하는 기법으로 특정 시간대의 버전 넘버를 보유하며, 변경 상태 값 True인 모든 레코드 추출하는 등 정교한 쿼리 생성에 활용해 개발 유연성 제공

마. Triggers on Tables : **사전에 등록(Subscribe)된 다수 대상 시스템(Target)**에 변경 데이터를 배포(Publish)하는 형태로 CDC 구현하는 기법, 시스템 관리 복잡도 증가, 변경 관리 어려우며 확장성 감소시키는 등 시스템 유지보수성 저하

바. Event Programming : 데이터 변경 식별 기능을 **어플리케이션**에 구현, 어플리케이션 개발 부담과 복잡도가 있으나 다양한 조건에 의해 CDC 구현 가능

사. Log Scanner on Database : DBMS 데이터에 대한 변경 여부와 변경 값 시간 등을 **트랜잭션 로그** 기록, 관리하는 기능, 이 로그에 대한 스캐닝 및 변경 내역에 대한 해석을 통해 CDC 구현 but 각 DBMS에 따라 트랜잭션 로그 관리 메커니즘이 상이해 다수의 이기종 데이터베이스를 사용하는 환경에서 적용 시 주의 필요(장점 : 데이터 베이스/데이터베이스 사용 애플리케이션에 대한 영향도 최소화, 변경 식별 지연시간 최소화, 트랜잭션 무결성에 대한 영향도 최소화, 데이터베이스 스키마 변경 불필요)

제3절 EAI(Enterprise Application Integration)

1. EAI(Enterprise Application Integration) 개요 : 기업 정보 시스템들의 데이터를 연계, 통합하는 소프트웨어 및 정보 시스템 아키텍처 프레임워크, 기업 간 이질적 정보 시스템들의 데이터를 연계함으로써 상호 융화 및 동기화되어 동작한다. EAI를 통해 다수의 정보 시스템에게 기업 내 주요한 데이터 엔티티들에 대한 폭넓고 통합적인 뷰 제공 가능, 이를 통해 비즈니스 프로세스 자동화하고 실시간으로 통합, 연계 가능(기존 단위 업무 위주의 정보 시스템 개발 시 필요에 따라 정보 시스템들 간 포인트 투 포인트 방식으로 데이터 연계하여 복잡, 정보 시스템 간 데이터 통합 어렵고 표준화 불가, 유지보수 및 관리 비용 상승)
• 포인트 투 포인트 방식으로 정보 시스템 개발 시 N개의 연결 대상 노드들이 존재할 경우 연결은 N(N-1)/2개 발생→비용 증가 극복 위해 'Hub and Spoke' 방식의 EAI 기반 구조 적용(가운데 지점에 허브 역할을 하는 브로커 두고 연결 대상 노드들의 데이터 연계 요구를 중계하여 발생 연결 개수 및 구조 단순화(각 연결 대상 노드들은 스포크에 해당
• EAI 구성 요소로서 정보 시스템과 EAI 허브(Engine) 간 연결성 확보 위한 어댑터(Adapter)가 있으며, 이 어댑터들을 매개로 연결된 각 정보 시스템들 간의 데이터 연동 경로 버스(Bus), 데이터 연동 규칙 통제하는 브로커(Broker), 데이터 형식 변환 담당하는 트랜스포머(Transformer) 존재
2. EAI 구현 유형
가. Mediation(intra-communication) : EAI 엔진이 중개자(Broker)로 동작, 유의미한 이벤트 발생을 식별해 사전 약속된 정보 시스템들에게 그 내용 전달 Publish / Subscribe Model
나. Federation(inter-communication) : EAI 엔진이 외부(고객, 파트너) 정보 시스템들로부터 데이터 요청들을 일괄적으로 수령해 필요한 데이터 전달 Request / Reply Model
3. EAI 기대 효과 : 정보 시스템 개발 및 유지보수 비용 절감, 지속적 발전 기반 확보, 협력사/파트너/고객과의 상호 협력 프로세스 연계 발전 기반 확보, 인터넷 비즈니스 기본 토대

제4절 데이터 연계 및 통합 기법 요약

1. 데이터 연계 및 통합 유형(동기화 기준) : 데이터 연계 및 통합 시 일괄(Batch) 작업 또는 비동기식 근접 실시간(Near Real Time) 방식 혼용 사용 가능(일괄 작업 시 대용량 데이터 처리 가능, 실시간 통합 시 관심 대상 영역 상태에 대한 빠른 파악 및 대응 가능)
• 일괄 작업 사례로 ETL 기능 통해 운영 시스템으로부터 **정기적, 반복적**으로 대량의 데이터를 획득해 ODS 구성하고, 데이터 웨어하우스나 데이터 마트 구성한 뒤 OLAP 정형/비정형 질의 통해 경영 분석 수행하는 작업
• 동기식 실시간 데이터 통합 사례로는 컨테이너 터미널, 공장 등의 생산, 운송 장비에 설치된 각종 센서들로부터 데이터를 실시간 획득해 운영 상태 모니터링하는 경우
• 데이터 중복을 허용하는 분산 저장 환경 구성을 통한 높은 확장성을 확보하는 빅데이터 저장 인프라스트럭처의 활용과 병행 설계되는 사례 등장
• 전통적인 ETL 기술은 데이터 웨어하우스 구성만을 주목적으로 했으나 최근 ODS, BI 플랫폼, MDM 허브, 하둡 클라우드 환경 등 다양한 데이터 통합 메커니즘 지원

일괄(Batch) 통합	비동기식 실시간 통합	동기식 실시간 통합
• 비실시간 데이터 통합 • 대용량 데이터 대상 • 높은 데이터 조작 복잡성 • 데이터 추출/변형/적재 • CDC • 감사 증적 • 웹 서비스/SOA • 교차 참조 • 데이터 재처리 허용 • 점대점 데이터 연계 • 자동화 도구 및 자체 개발 SW 혼용	• 근접 실시간 데이터 통합 • 중간 용량 데이터 • 중간 데이터 조작 복잡성 • 데이터 추출/변형/적재 • CDC • Data pooling and DA Streams • 웹 서비스/SOA • 감사 증적(audit trail) • 교차 참조 • 다수 데이터 원천 및 목표 시스템 • 데이터 재처리 허용 • 자동화 도구 자체 개발 SW 혼용	• 실시간 데이터 통합 • 목표 시스템 데이터 처리 가능 시에만 원천 데이터 획득 • 데이터 추출/변형/적재 • 웹 서비스/SOM • Single transaction integrations • 단일 트랜잭션 단위 데이터 통합 • 데이터 재처리 불가 • 단일 또는 다수 데이터 원천 • 감사 증적

• 최근 의사 결정 지원을 위해 전자메일, 각종 문서 파일 등에 보관되는 비정형/준정형 데이터의 중요성 부각

구분	전통적 데이터 처리 기법	빅데이터 처리 기법	비고
추출	• 운영 DB→ODS • ODS→데이터 웨어하우스	빅데이터 환경 →빅데이터 환경	특정 소스에서 타깃으로 데이터를 옮긴다는 측면 동일
변환	O	O	

로딩(적재)	O	O	
시각화	X	O	시각화를 통해 대용량 데이터에서 통찰력(Insight)을 획득하고자 하는 시도는 빅데이터의 고유한 특성
분석	• OLAP • 통계, 데이터 마이닝 기술	통계 데이터 마이닝 기술	각종 통계 도구, 기법과 데이터 마이닝의 분석 모델 설계
리포팅	비즈니스 인텔리전스	비즈니스 인텔리전스	
인프라 스트럭처	• SQL • 전통적 RDBS 인스턴스 (HA 포함)	• NoSQL 등 • 초대형 분산 (Redundant) 데이터 스토리지	전통적 데이터 저장 메커니즘 대비 매우 다수의 노드에 중복을 허용하는 방식으로 데이터를 저장하는 것은 빅데이터의 고유한 특성

제5절 대용량 비정형 데이터 처리

1. 대용량 로그 데이터 수집 : 로그는 기업에서 발생하는 대표적인 비정형 데이터다. 과거에는 시스템의 문제 상황을 보존, 서비스 접근/사용 로그를 기록하는 용도로 사용했으나 최근에는 사용자 행태 분석 등 기업의 주요 비즈니스 영역인 마케팅과 영업 전략에 필수적인 정보 생성하는데 사용, 로그 분석하기 위해서는 성능과 확장성의 시스템 필요

- 대용량 비정형 데이터 수집 시스템 특징

가. 초고속 수집 성능과 확장성 : 실시간으로 발생하는 대용량 데이터를 놓치지 않고 수집할 수 있어야 한다. 수집 대상 서버가 증가하면 그만큼 에이전트 수 늘리는 확장 구조 갖는다.

나. 데이터 전송 보장 메커니즘 : 수집한 데이터는 처리, 분석을 위한 저장소인 분산 파일시스템이나 데이터베이스, NoSQL 등에 저장된다. 데이터의 종류에 따라 수집에서 저장소까지의 양 종단점 간 데이터 전송 안정성 수준 제어, 단계별로 전송될 때마다 신호를 주고 받아서 이벤트 유실되지 않도록 하거나 여러 단계의 데이터 전송에서 인접한 단계로만 데이터 전송 보장하는 방법이 있다.(성능과 안정성 trade off 관계)

다. 다양한 수집과 저장 플러그인 : 로그나 성능 데이터 수집할 수 있는 기본 기능 뿐만 아니라 일부 외부(소셜) 서비스로부터 몇 가지 설정만으로 데이터를 수집할 수 있도록 내장 플러그인을 제공해야한다.

라. 인터페이스 상속을 통한 애플리케이션 기능 확장 : 서비스에 적용하는 과정에서 수집 시스템에서 제공하는 기능을 사용하지만 업무 특성 상 일부 기능 수정해야할 때 인터페이스를 확장해 원하는 부분만 비즈니스 용도에 맞게 수정할 수 있어야 한다. 오픈소스 데이터 수집 시스템인 Flume-NG의 경우 비정형 데이터의 주요 특징을 포함하고 있으며 빅데이터 플랫폼을 구축할 때 수집 시스템으로 많이 활용된다.

① Application Tier 애플리케이션 단계 : 데이터 발생 ② Flume Collector Tier : 데이터 수집 ③ Flume Storage Tier : 저장 ④ Data Tier : 데이터 저장소(하둡) 보관

2. 대규모 분산 병렬 처리 : 대규모 컴퓨팅과 연산 작업이 필요하다면 하둡 사용(하둡은 대규모 분산 병렬 처리의 업계 표준인 맵리듀스 시스템과 분산 파일시스템인 HDFS로 구성된 플랫폼 기술)

- 하둡 특징

• **선형적인 성능과 용량 확장** : 하둡 구축은 여러 대의 서버로 클러스터 만드는 것,으로 하둡은 필요 시 서버를 추가하면 연산 기능과 저장 기능이 서버의 대수에 비례하여 증가한다.(하둡이 비공유 분산 아키텍처 시스템이기 때문) 확장성 뛰어나며 선형적 성능 확장 가능

• **고장 감내성** : 데이터는 3중 복제되어 서로 다른 물리서버에 저장된다. 장애가 발생하더라도 데이터 유실 방지, 관리자의 개입 없이 시스템 수준에서 자동으로 동작

• **핵심 비즈니스 로직에 집중** : 비즈니스 목적에 맞게 간단한 코드만 작성해주면 장애 상황 발생하더라도 자동 복구(failover) 기능이 있어 비즈니스 로직에만 집중할 수 있도록 내부적으로 최적화하여 처리

• 풍부한 에코시스템 형성 : 비즈니스의 요구 사항을 위해서는 필연적으로 하둡 같은 인프라 시스템에서 동작하는 다양한 응용 기술 필요 데이터 수집 기술로 Flume-NG, 데이터 연동 기술로 Sqoop, 데이터베이스 기술 NoSQL로 HBase, 대용량 SQL 질의 기술로 Hive와 Pig, 실시간 질의 기술로 Impala, Tajo, 워크플로 관리 기술로 Oozie, Azkaban 등이 있다.

3. 데이터 연동 : 비정형 데이터 분석의 경우 기업 내의 축적된 데이터와 정보 연동이 필요하다. 기간 계 시스템이 데이터베이스를 대상으로 맵리듀스와 같은 대규모 분산 병렬 처리를 하는 것은 심한 부하를 야기할 수 있어 정형/비정형 데이터 간의 연계 분석을 위해 데이터베이스의 데이터를 하둡으로 복사한 후 하둡에서 대규모 분산 병렬 처리 수행한다. 그 결과로 생성된 요약된 데이터셋을 다시 데이터베이스에 기록하는 식으로 작업→대표적인 오픈 소스 기반의 솔루션 스쿱(Sqoop)

- 스쿱을 이용해서 데이터베이스로부터 하둡으로 데이터 전송하는 스크립트
• 데이터를 가져올 데이터베이스 접속 정보 입력
• 가져올 데이터에 대한 SQL 입력
• 동시에 몇 개의 프로세스 실행하여 데이터 가져올지 지정
• 데이터베이스의 키 칼럼 입력
• 데이터베이스로부터 가져온 데이터 저장할 하둡 상의 경로 지정

4. 대용량 질의 기술 : 하둡은 저비용으로 대용량 데이터를 저장하고 빠르게 처리할 수 있는 시스템이며 빅데이터 구현을 위한 혁신적인 플랫폼 기술로 사용되고 있다. 하지만 여전히 코딩이 필요하여 어려움→하이브(Hive) 등장 : 사용자에게 친숙한 SQL 질의 기술 이용하여 하둡 상의 저장된 데이터를 쉽게 처리하고 분석해주는 도구 and SQL on 하둡 : 실시간 SQL 질의 분석 기술

• 아파치 드릴(Drill) : 맵알(MapR)이 주축이 되어 진행하는 프로젝트, 드레멜 아키텍처와 기능 동일하게 구현한 오픈 소스 버전 드레멜
• 아파치 스팅거(Stinger) : 호튼웍스에서 개발 주도, 기존의 하이브 코드 이용하여 성능 개선
• 샤크(Shark) : 인메모리 기반의 대용량 데이터웨어하우징 시스템, 하이브와 호환
• 아파치 타조(Tajo) : 고려대외 국내 회사인 그루터(Gruter) 합류하여 개발 진행

• 호크(HAWQ) : 피보탈에서 개발한 프로젝트, 상용과 커뮤니티 2가지 버전
• 프레스토 : 페이스북에서 자체 개발하여 사용하고 있는 하둡 기반 데이터웨어하우징 엔진
• 임팔라(Impala) : 클라우데라 개발 주도

- 임팔라의 구성 요소
• 클라이언트 : 임팔라에 접속하여 테이블 관리, 조회 등의 작업 수행
• 메타스토어 : 임팔라로 분석할 대상 데이터들의 정보 관리, 하이브의 메타데이터 같이 사용
• 임팔라 데몬 : 시스템에서는 ImpalaD로 표시되며 클라이언트의 SQL 질의 받아 데이터 파일들의 읽기/쓰기 작업 수행한다. 내부적으로 질의 실행계획기, 질의 코디네이터, 질의 실행엔진으로 구성된다.
• 스테이트스토어 : 임팔라 데몬들의 상태 체크하고 건강 정보 관리해주는 데몬, 장애가 생겼을 경우 이 사실을 알려서 장애가 발생한 데몬에게는 질의 요청이 가지 않도록 해준다.
• 스토리지 : 분석할 데이터를 저장하는 에이치베이스, 하둡분산파일시스템 지원

- 모든 노드에 임팔라 데몬이 구동되며 사용자는 이 데몬들의 구동된 임의 노드에 JDBC나 ODBC 또는 임팔라쉘을 이용하여 질의 요청할 수 있다. 그러면 사용자 질의는 데이터의 지역성 고려하여 노드 전체로 분산되어 수행된다. 사용자의 질의 요청을 받은 코디네이터 데몬은 분산되어 수행된 각 임팔라 노드들의 부분 결과들을 취합하여 결과 값을 만들어 사용자에게 제공(실제 운영 환경에서는 **라운드로빈 방식**으로 사용자 질의를 분산시켜서 질의에 대해 전 노드들이 코디네이터 역할 고르게 수행하도록 한다.)

- 하이브가 하둡에 저장된 다양한 형태의 대용량 비정형 데이터 효율적으로 처리하는 표준 SQL 솔루션으로 사용되고 있지만, 더 빠른 처리가 필요하기 때문에 임팔라 기술 대두

제2장 데이터 처리 기술

제1절 분산 데이터 저장 기술 : 분산 데이터 저장 기술은 네트워크 상에서 데이터를 저장, 조회, 관리할 수 있으며 저장 데이터의 정형화 여부와 데이터 모델에 따라 분산 파일시스템과 클러스터 데이터베이스, Key-Value 저장소 정도로 구분할 수 있다. 기존의 아키텍처는 고가의 마스터 서버에서 많은 역할을 수행하는 중앙 집중 방식이었다. 이는 장애 발생 자체를 차단하기 위해 노력한 구조인 반면 GFS, BigTable 같은 플랫폼은 저가의 PC급 서버들로 클러스터를 구성해 마스터서버의 역할을 축소했으며, 장애 발생을 염두에 두고 디자인됐다. 이러한 아키텍처는 대용량 데이터와 대규모 확장성, 소유 총비용 절감이 특징이다.

1. 분산 파일 시스템(DFS) : 비대칭형(asymmetric) 클러스터 파일 시스템은 성능과 확장성, 가용성 면에서 적합한 분산 파일 시스템 구조로 파일 메타데이터를 관리하는 전용 서버를 별도로 둠으로써 메타데이터에 접근하는 경로와 데이터에 접근하는 경로를 분리해 이를 통하여 파일 입출력 성능을 높이면서 독립적인 확장과 안전한 파일 서비스를 제공한다. But 메타데이터 서버에 부하가 집중 될 수 있고 single-of-failure 지점이 되는 문제점 내포

가. 구글 파일 시스템(Google File System)
- 설계 시 가정
• 저가형 서버로 구성된 환경으로 서버의 고장 빈번히 발생
• 대부분의 파일은 대용량
• 연속적으로 많은 데이터를 읽는 연산 또는 임의의 영역에서 적은 데이터를 읽는 연산은 작업이 부하된다.
• 쓰기 연산은 주로 순차적으로 데이터 추가하며, 파일에 대한 갱신은 드물게 이루어진다.
• 여러 클라이언트에서 동일한 파일 데이터 추가하는 환경에서 동기화 오버헤드 최소화할 수 있는 방법 요구
• 낮은 응답 지연시간보다 높은 처리율이 더 중요
- GFS의 클라이언트는 파일 시스템 인터페이스와 유사한 자체 인터페이스를 지원한다. 또한 여러 클라이언트에서 원자적인 데이터 추가(atomic append) 연산을 지원하기 위한 인터페이스 지원한다.
- GFS에서 파일은 고정된 크기의 chunk들로 나누어 여러 복제본과 함께chunk 서버들에 분산/저장된다. 클라이언트는 파일에 접근하기 위하여 마스터로부터 해당 파일의 chunk가 저장된 chunk 서버의 위치와 핸들을 먼저 받아 온다.
- 직접 chunk 서버로 파일 데이터 요청한다. GFS 마스터는 단일 마스터 구조로 파일 시스템 이름 공간과 파일의 chunk 매핑 정보, 각 chunk가 저장된 chunk 서버들의 위치 정보 등 모든 메타데이터를 메모리상에서 관리한다. GFS에서는 기본 chunk 크기를 64MB로 지정하여 파일 메타데이터의 크기를 줄이며 해시 테이블 구조 등을 사용하여 메모리 상에서 보다 효율적인 메타데이터 처리를 지원한다. 마스터에서는 주기적으로 하트비트(heartbeat) 메시지를 이용하여 chunk 서버에 저장된 chunk들의 상태를 체크하며 chunk 재복제, 재분산 등의 회복 동작 수행한다.
- 마스터에 대한 장애 처리를 위해 파일시스템 이름 공간과 파일의 chunk 매핑 변경 연산을 로깅하고 마스터의 상태를 여러 새도 마스터에 복제한다.
- chunk 서버는 로컬 디스크에 chunk를 저장, 관리하면서 클라이언트로부터의 chunk 입출력 요청을 처리한다. Chunk는 마스터에 의해 생성/삭제될 수 있으며 유일한 식별자에 의해 구별된다. 마스터는 하나의 chunk 서버를 primary로 지정하여 복제본 갱신 연산을 일관되게 처리할 수 있도록 보장

나. 하둡 분산 파일 시스템(HDFS) : 구글 파일 시스템과 아키텍처와 사상을 그대로 구현한 클로닝(Cloning) 프로젝트, HDFS는 하나의 네임노드와 다수의 데이터노드로 구성된다. 네임노드는 파일 시스템의 이름 공간을 관리하면서 클라이언트로부터의 파일 접근 요청을 처리한다. HDFS에서 파일 데이터는 블록 단위로 나뉘어 여러 데이터노드에 분산, 저장된다.(블록들은 가용성 보장을 위해 다시 복제, 저장된다.) 네임노드는 데이터 노드들로부터 하트비트(Heartbeat)를 주기적으로 받으면서 데이터노드들의 상태를 체크한다. 클라이언트, 네임노드, 데이터노드 간의 통신을 위해 TCP/IP 네트워크상에서 RPC(Remote Prodedure Call) 사용

다. 러스터(Luster) : 객체 기반 클러스터 파일 시스템으로 클라이언트 파일 시스템, 메타데이터 서버, 객체 저장 서버들로 구성되며 이들은 고속 네트워크로 연결된다. 러스터에서는 계층화된 모듈 구조로 TCP/IP, 인피니밴드(Infiniband), 미리넷(Myrinet)과 같은 네트워크 지원한다.
- 클라이언트 파일 시스템은 리눅스 VFS(Virtual File System)에서 설치할 수 있는 파일 시스템으로 메타데이터 서버와 객체 저장 서버들과 통신하면서 클라이언트 응용에 파일 시스템 인터페이스를 제공한다.

- 메타데이터 서버는 파일 시스템의 이름 공간과 파일에 대한 메타데이터를 관리하며 객체 저장 서버는 파일데이터를 저장하고 클라이언트로부터의 객체 입출력 요청을 처리한다. 객체는 객체 저장 서버들에 스트라이핑되어 분산, 저장된다.
- 러스터는 유닉스(Unix) 시맨틱을 제공하면서 파일 메타데이터에 대해서는 라이트백 캐시(Write Back Cache)를 지원한다. 이를 위해 클라이언트에서 메타데이터 변경에 대한 갱신 레코드를 생성하고 나중에 메타데이터 서버에 전달한다. 메타데이터 서버는 전달된 갱신 레코드를 재수행하여 변경된 메타데이터를 반영한다. 또한 메타데이터를 동시에 접근하는 부하에 따라 클라이언트 캐시에서 라이트백 캐시를 지원하거나 메타데이터 서버에서 메타데이터를 처리하는 방식 적용한다.
- 러스터는 메타데이터 서버에서 처리하도록 하는 방식을 사용해 메타데이터에 대한 동시 접근이 적으면 클라이언트 캐시를 이용한 라이트백 캐시 사용, 많으면 클라이언트 캐시를 사용하여 오버헤드 줄인다.
- 동시성 제어 위해 별도의 잠금 사용 : 메타데이터에 접근하기 위해서는 메타데이터 서버의 잠금 서버로부터 잠금을 획득해야 한다. 파일 데이터를 접근하기 위해서는 해당 데이터가 저장된 객체 저장 서버의 잠금 서버로부터 잠금을 획득해야 한다.
- 인텐트(Intent) 기반 잠금 프로토콜 : 클라이언트와 메타데이터 서버 간의 네트워크 트래픽을 최소화하기 위하여 메타데이터에 대한 잠금 요청 시 메타데이터 접근 의도를 같이 전달

구분	GFS	하둡 DFS(HDFS)	러스터
Open Source	지원	지원	지원
Chunk based	지원	지원	지원 안함
Support Replication	지원	지원	지원 안함
Multiple metadata server supported	지원 안함	지원 안함	지원 안함
Locks used to maintain atomicity	지원	지원	지원
Uses a DB for storing metadata	지원 안함	지원 안함	지원 안함
Adding nodes without shutting down the system	지원	지원	지원
POSIX support	지원 안함	지원 안함	지원
Supports file modification	지원 안함 (append는 지원)	지원 안함	지원

2. 데이터베이스 클러스터 : 데이터 통합 시 성능/가용성 향상 위해 데이터베이스 차원의 파티셔닝 또는 클러스터링을 이용한다. 혜택 : 병렬 처리 통해 빠른 데이터 검색, 처리 성능 획득, 성능의 선형적인 증가 효과, 장애 발생 시 서비스 중단되지 않는 고가용성 확보
- 데이터베이스 파티셔닝은 데이터베이스 시스템을 구성하는 형태에 따라 단일 서버 내의 파티셔닝과 다중 서버 사이의 파티셔닝으로 구분 가능, 리소스 공유 관점에서는 공유 디스크(Shared Disk)와 무공유 디스코(Shared Nothing)로 구분
• 무공유 : 무공유 클러스터에서 각 데이터베이스 인스턴스는 데이터 파일을 자신의 로컬 디스크에 저장하며 이 파일들은 노드 간 공유하지 않는다. 인스턴스나 노드는 완전히 분리된 데이터의 서브 집합에 대한 소유권을 가지고 있으며 각 데이터는 소유권을 갖고 있는 인스턴스가 처리한다. 장점 : 노드 확장 무제한, 단점 : 각 노드 장애 발생 시를 대비해 별도의 폴트톨러런스(Fault-Tolerance)를 구성해야 한다.
• 공유 디스크(ex Oracle RAC) : 클러스터에서 데이터 파일을 논리적으로 모든 데이터베이스 인스턴스 노드들과 공유하며 각 인스턴스는 모든 데이터에 접근할 수 있다. 데이터를 공유하려면 SAN(Storage Area Network)과 같은 공유 디스크가 반드시 있어야 하며, 모든 데이터를 수정할 수 있기 때문에 노드 간의 동기화 작업 수행을 위한 별도 커뮤니케이션 채널 필요하다. 장점 : 높은 수준의 폴트톨러런스 제공(하나의 노드만 살아있어도 서비스 가능), 단점 : 클러스터가 커지면 디스크 영역에서 병목현상 발생

가. **Oracle RAC 데이터베이스 서버** : 4노드 RAC 구성 모델로 클러스터의 모든 노드에서 실행되며 데이터는 **공유** 스토리지에 저장된다. 모든 테이블에 동등하게 액세스하며 특정 노드가 데이터를 소유하는 개념이 없다. 따라서 데이터를 파티셔닝할 필요는 없지만 성능 향상을 위해 파티셔닝한다.
• 가용성 : 한 노드에서 장애 발생 시 Oracle RAC는 나머지 노드에서 계속 실행된다. 모든 응용 프로그램(사용자)은 투명하게 다시 연결되어 클러스터의 나머지 노드에 분산된다.
• 확장성 : 추가 처리 성능이 필요하면 응용 프로그램이나 데이터베이스를 수정할 필요 없이 새 노드를 클러스터에 쉽게 추가할 수 있다. 클러스터의 모든 노드 간 균형 유지되도록 로드가 다시 분산(최대 100개 노드 지원)
• 비용 절감 : 표준화된 소규모(CPU 4개 미만) 저가형 상용 하드웨어의 클러스터에서도 고가의 SMP 시스템만큼 효율적으로 응용 프로그램 실행하여 하드웨어 비용 절감한다. 일반적으로 4노드 이상 잘 구성하지 않는다. 도입 비용 때문에 확장성이 중요한 데이터보다는 고가용성을 요구하는 데이터에 많이 사용한다.

나. **IBM DB2 ICE(Integrated Cluster Environment)** : DB2는 CPU, 메모리, 디스크를 파티션별로 독립적으로 운영하는 **무공유** 방식의 클러스터링을 지원한다. 애플리케이션은 여러 파티션에 분산된 데이터베이스를 하나의 데이터베이스(Single View Database)로 보게 되고, 데이터가 어느 파티션에 존재하고 있는지 알 필요가 없다. 데이터와 사용자가 증가하면 애플리케이션의 수정 없이 기존 시스템에 노드 추가하고 데이터 재분배하여 시스템 성능과 용량 일정하게 유지할 수 있다. 각 노드로 분산되는 파티셔닝 구성에 따라 성능 차이가 발생하며 장애 발생 시 해당 노드에서 서비스하는 데이터에 대한 별도의 페일오버(failover) 메커니즘 필요하다. 따라서 DB2 이용하여 클러스터 구성 시 가용성 보장을 위해 공유 디스크 방식을 이용한다. 공유 디스크에 저장된 데이터 파일에 대해 특정 시점에서는 특정 노드에 의해 서비스되지만 장애 발생 시 다른 노드가 해당 데이터 서비스 처리하는 방식으로 가용성 보장

다. **마이크로소프트 SQL Server** : 연합(Federated) 데이터베이스 형태로 여러 노드로 확장할 수 있는 기능 제공한다. 연합 데이터베이스는 **디스크를 공유하지 않는** 독립된 서버에서 실행되는 서로 다른 데이터베이스들 간의 논리적 결합이며, 네트워크를 이용하여 연결된다. 데이터는 관련된 서버들로 수평적으로 분할된다. 테이블을 논리적으로 분리해 물리적으로는 분산된 각 노드에 생성하고 각 노드의 데이터베이스 인스턴스 사이에 링크를 구성한 후 모든 파티션에 대해 UNION ALL을 이용해 논리적인 뷰(VIEW) 구성하는 방식으로 분산 환경의 데이터에 대한 싱글 뷰 제공한다.→DVP(Distributed Partitioned View) 단점 : DBA나 개발자가 파티셔닝 정책에 맞게 테이블과 뷰를 생성해야 하며 전역 시스카(Global Schema) 정보가 없어 질의 수행을 위해 모든 노드를 액세스해야 한다. 노드가 많아지거나 추가/삭제가 발생한 경우 파티션을 새로 해야한다. 페일오버 별도 구성(SQL Server에서도 페일오버 메커니즘 제공하지만, Active-Standby 방법 사용)

라. **MySQL** : **무공유** 구조에서 메모리(디스크) 기반 데이터베이스의 클러스터링을 지원하며 특정한 하드웨어 및 소프트웨어 요구하지 않고 병렬 서버구조로 확장 가능하다.
- 관리 노드 : 클러스터 시작과 재구성 시에만 관여
- 데이터 노드 : 데이터 저장하는 노드
- MySQL 노드 : 클러스터 데이터에 접근 지원하는 노드
- 가용성 높이기 위해 데이터를 다른 노드에 복제하여 특정 노드 장애 발생하더라도 지속적인 데이터 서비스 가능하다. 장애가 났던 노드가 복구되어 클러스터에 투입된 경우에도 기존 데이터와 변경된 데이터에 대한 동기화 작업이 자동을 수행된다. 데이터는 동기화 방식으로 복제되며 이를 위해 데이터 노드 간에 별도의 네트워크 구성한다.
- 최근 디스크 기반의 클러스터링 제공한다. 인덱스가 생성된 칼럼은 기존과 동일하게 메모리에 유지되지만 인덱스 생성하지 않은 칼럼은 디스크에 저장된다. 디스크에 저장된 데이터와 JOIN 연산을 수행할 경우 성능이 좋지 않기 때문에 애플리케이션 개발 시 주의해야 한다. 인덱스로 구성된 칼럼은 메모리에 있으므로 데이터의 크기와 메모리의 크기를 고려하여 인덱스 생성과 클러스터의 참여하는 장비의 메모리 산정해야 한다.
- MySQL 클러스터 구성 시 제한 사항
• 파티셔닝은 Linear Key 파티셔닝만 사용 가능
• 클러스터에 참여하는 노드(SQL 노드, 데이터 노드, 매니저 포함) 수 255 제한(데이터 노드는 최대 48까지 가능)
• 트랜잭션 수행 중 롤백을 지원하지 않으므로 문제 발생 시 전체 트랜잭션 이전으로 롤백
• 하나의 트랜잭션에 많은 데이터를 처리하는 경우 여러 개의 트랜잭션으로 분리하여 처리
• 칼럼명 길이는 31자, 데이터베이스와 테이블명 길이는 122자 제한하며 메타데이터(속성정보)는 20,320개까지만 가능
• 클러스터에서 생성할 수 있는 테이블 수 최대 20,320개이며 테이블의 키는 32개 최대
• 모든 클러스터의 기종은 동일해야 한다.
• 운영 중에 노드 추가/삭제 불가능
• 디스크 기반 클러스터인 경우 tablespace의 개수는 2^32개, tablespace당 데이터 파일의 개수는 2^16개, 데이터 파일의 크기는 32GB까지 가능

3. NoSQL : NoSQL은 Key와 Vale의 형태로 자료를 저장하고 빠르게 조회할 수 있는 자료 구조를 제공하는 저장소다. 복잡한 Join 연산 기능은 지원하지 않지만 대용량 데이터와 대규모 확장성을 제공한다.

가. **구글 빅테이블** : 구글 내부에서 사용하는 데이터 저장소다. AppEngine에서 사용하는 데이터 저장소가 빅테이블이다.
- 데이터 모델 : 빅테이블은 multi-dimension sorted hash map을 파티션하여 분산 저장하는 저장소다. 테이블 내의 모든 데이터는 row-key의 사전적 순서로 정렬, 저장된다. Row는 n개의 column-family를 가질 수 있으며 column-family에는 column-key, value, timestamp의 형태로 저장된다. 하나의 row-key, column-family 내에 저장된 데이터는 column-key의 사전적 순서로 정렬돼 있다. 동일한 column-key에 대해 타임스탬프가 다른 여러 버전의 값이 존재할 수 있다. 따라서 BigTable에 저장되는 하나의 데이터(map)의 키 값 또는 정렬 기준은 rowkey+columnkey+timestamp가 된다. 테이블의 파티션은 rowkey를 이용하며 분리된 파티션은 분산된 노드에서 서비스하도록 한다. 분리된 파티션을 Tablet이라 하며 하나당 크기는 보통 100~200MB다.
- 페일오버 : 장애 발생 시 빅테이블의 마스터는 장애가 발생한 노드에서 서비스되던 Tablet을 다른 노드로 재할당시킨다. 재할당 받은 노드는 초기화 작업을 수행한 후 데이터 서비스를 한다. 빅테이블은 데이터베이스 클러스터 분류로 나누자면 **공유** 디스크 방식이다. 빅테이블의 SPOF(Single Point Of Failure)는 마스터다. 빅테이블은 분산 락 서비스를 제공하는 Chubby를 이용해 마스터를 계속 모니터링하다가 마스터에 장애가 발생하면 가용한 노드에 마스터 역할을 수행하도록 한다. Chubby는 자체적으로 폴트톨러런스 지원 구조이기 때문에 절대로 장애가 발생하지 않는다. 빅테이블은 파일시스템, Map&Reduce 컴퓨팅 클러스터와 동일한 클러스터 위에 구성된다. 실시간 서비스 뿐만 아니라 대용량 데이터 분석 처리에 적합하도록 구성됐다.
- AppEngine : AppEngine 내에서 운영되 애플리케이션의 데이터 저장소를 제공하며 내부적으로는 빅테이블을 이용한다. 사용자에게 직접 빅테이블의 API 공개하지 않고 추상 계층을 두고 데이터 모델에 대해서도 추상화되어 있다. 사용자 테이블을 생성할 경우 빅테이블의 특정 테이블의 한 영역만을 차지하게 된다. 빅테이블에서는 별도의 사용자 정의 인덱스 제공하지 않고, ApppEngine에서는 사용자가 수행하는 쿼리를 분석해 자동으로 인덱스 생성

나. **아마존 SimpleDB** : 웹 애플리케이션에서 사용하는 데이터의 실시간 처리 지원한다. 주로 아마존의 다른 플랫폼 서비스와 같이 사용된다. 아마존 내부 서비스 간 네트워크 트래픽은 무료이고 외부와의 In/Out 트래픽에는 요금을 부과하는 아마존 서비스의 가격 정책 때문이다. 사용자는 EC2에서 수행되는 웹 서버로 접근하고 웹 서버에서 SimpleDB의 데이터 조회해 적절하게 가공한 후 사용자에게 제공하는 형태로 구성된다. SimpleDB는 하나의 데이터에 대해 여러 개의 복제본을 유지하는 방식으로 가용성을 높인다. SImpleDB에서는 Eventual Consistency 정책(트랜잭션 종료 후 데이터는 모든 노드에 즉시 반영되지 않고 초 단위로 지연되어 동기화)을 취한다. SimpleDB는 관계형 데이터 모델과 표준 SQL 지원하지 않으며 전용 쿼리 언어 이용하여 데이터 조회한다. Domain, Item, Attribute, Value로 구성되며 스키마가 없는 구조다.
- 도메인 : 관계형 데이터베이스의 테이블과 동일한 개념, 하나의 도메인에는 최대 10GB 데이터 저장 가능, 사용자는 100개의 도메인 가질 수 있다.
- Items : 관계형 데이터베이스의 레코드와 동일한 개념, 독립적인 객체 나타내며 하나 이상의 Attribute 가진다. 한 item은 최대 256개의 Attribute 가질 수 있다.
- Attribute : 관계형 데이터베이스의 칼럼과 동일한 개념, 사용하기 전 미리 정의할 필요 없다. Name, Value 쌍으로 데이터 저장하고 저장되는 데이터의 Name이 Attribute의 이름이 된다. Item의 특정 Attribute(Cell)에는 여러 개의 값을 저장할 수 있다.
• 한번에 하나의 도메인에 대해서만 쿼리를 수행해야 한다. 1+N(mast-slave) 관계의 데이터 모델을 갖는 두 개의 도메인으로부터 데이터를 조회할 경우 쿼리가 여러 번 수행돼야 하는 단점이 있다. 클라이언트는 SOAP 또는 REST 프로토콜 이용하여 SimpleDB 이용할 수 있으며 다음과 같은 API 제공한다.

CreateDomain	도메인 생성
DeleteDomain	도메인 삭제
ListDomain	모든 도메인 목록 가져오기
PutAttruibutes	Item 생성, Attribute 값 추가
DeleteAttributes	Attribute 값 삭제

GetAttributes	Attribute 값 조회
Query	쿼리 이용하여 조건에 맞는 여러 item 조회하기(한번의 쿼리는 최대 5초 이내에 수행되어야하며(5초 이상 timeout 발생), 쿼리 결과 받을 수 있는 item 최대 수는 256개

다. **마이크로소프트 SSDS** : 고가용성 보장, SSDS 데이터 모델은 컨테이너(테이블과 유사한 개념이지만 하나의 컨테이너에 여러 종류의 엔티티 저장 가능), 엔티티(레코드와 유사한 개념으로 하나의 엔티티는 여러 개의 property 가질 수 있으며 property는 name-value 쌍으로 저장된다.)로 구성돼 있다. 일반적으로 CustomerA의 주문정보(Order)와 주문상세정보(OrderDetail)를 저장하기 위해 Order 테이블과 OrderDetail 테이블을 생성한다. 하지만 SSDS에서는 CustomerA라는 Container 만들고 Order와 OrderDetail entity 생성한 컨테이너에 모두 저장한다. 즉 CUstomerId가 파티셔닝 키가 되고 파티셔닝 대상은 컨테이너가 된다. 이 컨테이너는 여러 노드에 분산, 관리된다. 쿼리는 하나의 컨테이너만을 대상으로 한다. 컨테이너의 생성/삭제, 엔티티의 생성/삭제, 조회, 쿼리 등의 API를 제공하고 SOAP/REST 기반의 프로토콜을 지원한다.

제2절 분산 컴퓨팅 기술

1. MapReduce : 분할정복 방식으로 대용량 데이터를 병렬로 처리할 수 있는 프로그래밍 모델이다. 특별한 옵션을 주지 않으면 Map Task 하나가 1개의 블록(64MB)을 대상으로 연산 수행한다. Map 과정에서 생산된 중간 결과물들을 Reduce Task들(사용자가 개수 지정)이 받아와서 정렬 및 필터링 작업 거쳐 최종 결과물 만들어 낸다.

가. **구글 MapReduce** : 병렬화, 장애 복구 등의 복잡성 추상화해 핵심 기능 구현에만 집중
- 프로그래밍 모델 : MapReduce는 Map과 Reduce 2단계로 나뉘며 Map에서는 Key와 Value 쌍들을 입력으로 받는다. 하나의 Key, Value 쌍은 사용자가 정의한 Map 함수 거쳐 다수의 새로운 Key, Value 쌍들로 변환되어 로컬 파일 시스템에 임시 저장된다. 임시 파일들은 프레임워크에 의해 Reduce에게 전송되는데 이 과정에서 자동으로 Shuffling과 Group by 정렬 한후 Reduce의 입력 레코드로 들어가는데 형식은 Key와 Value 리스트다. Reduce 입력 레코드들은 사용자가 정의한 Reduce 함수 통해 최종 Output 산출된다.(Map, Reduce 두 함수 작성만으로 대규모 병렬 연산 작업 수행 가능)
- 실행 과정 : 사용자가 MapReduce 프로그램을 작성해 실행하면 마스터는 사용자의 프로그램에서 지정한 입력 데이터소스를 가지고 스케줄링한다. 하나의 큰 파일은 여러 개의 파일 split들로 나뉘며 각 split들이 Map 프로세스들의 할당 단위가 된다. Split 단위는 블록 사이즈인 64MB나 128MB가 되며 split 수만큼 Map Task들이 워커로부터 fork됨과 동시에 실행되 Output을 로컬 파일 시스템에 저장한다. Output 값들은 Partitioner라는 Reduce 번호 할당해주는 클래스 통해 어떤 Reduce에게 보내질지 정해진다. 특별히 지정하지 않으면 Key 해시값을 Reduce 개수로 Modular 게산한 값이 부여되어 동일한 Key들은 같은 Reduce로 배정된다. Map 단계가 끝나면 원격 Reduce 워커들이 자기에 할당된 Map 중간값들을 네트워크로 가져, 사용자의 Reduce 로직을 실행해 최종 산출물을 얻어낸다. 보통 Reduce 개수는 Map 개수보다 적으며 Map 중간 데이터 사이즈에 따라 성능이 좌우된다. 분산 Grep이나 빈도 수 계산 등의 작업은 Map 단계 거치며 데이터 사이즈가 크게 줄고, 줄어든 크기만큼 Reduce 오버헤드도 줄어듦에 따라 성능상 이점 많다. 정렬 같은 작업은 오버헤드에 따른 수행 성능 저하되므로 MapReduce 모델이 적합하지 않다.
- 폴트톨러런스 : 각 프로세스에서는 Master에게 Task 진행 상태를 주기적으로 보낸다. 특정 워커의 Task 더 이상 진행되지 않거나 상태 정보를 일정 시간 동안 받지 못하면(Heartbeat Timeout) Task 문제있다고 결론을 내린다. 특정 Map이나 Reduce Task 죽은 경우 해당 Task가 처리해야할 데이터 정보만 다른 워커에게 전해주면 워커는 받은 데이터 정보를 인자로 새로운 Task 재실행(MapReduce는 무공유 아키텍처이므로 메커니즘 간단)

나. **Hadoop MapReduce** : 아파치 오픈소스 프로젝트로 자바 언어로 구현한 시스템
- 아키텍처 : 데몬 관점에서 하둡은 4개의 구성요소 가지고 있다. 네임노드와 데이터노드는 분산 파일 시스템의 데몬들이다. JobTracker는 MapReduce 시스템의 마스터고, TaskTracker는 워커 데몬이다. TaskTracker는 JobTracker에게 3초에 한번씩 주기적으로 Heartbeat 보내 살아 있다고 알린다. 클라이언트에서 하둡 작업을 실행하면 프로그램 바이너리와 입출력 디렉터리와 같은 환경 정보들이 JobTracker에게 전송된다. JobTracker에서는 작업을 다수의 Task로 쪼갠 후 그 Task들을 어떤 TaskTracker에게 보내면 데이터 지역성을 보장할지도 감안해 내부적으로 스케줄링하여 큐(Queue)에 저장한다. TaskTracker에서 Heartbeat 보내면 JobTracker는 먼저 해당 TaskTracker에게 할당된 Task가 있는지 큐에서 살펴본다. Task가 있으면 하트비트의 Response 메시지에 Task 정보를 실어서 TaskTracker에게 보낸다. TaskTracker는 Response 메시지의 내용 분석해 프로세스를 fork해 자기에게 할당된 Task 처리한다.
- 하둡의 성능 : MapReduce에서 Sort는 Map에서 Reduce로 넘어가는 과정에서 항상 발생하는 내부적인 프로세스다. Sort 작업은 데이터가 커질수록 처리 시간이 선형적으로 증가한다. 클러스터 구성 서버들의 숫자를 늘림으로써 처리 시간을 줄일 수 있는 것은 아니다. 플랫폼 자체적으로 선형 확장성을 갖고 있어야 처리 시간을 줄일 수 있다.
- 하둡 사용 현황 : WebMap은 알려진 웹 페이지들의 모든 edge 및 링크 정보를 계산해 그 결과를 다양한 검색 애플리케이션에서 사용할 수 있도록 해주는 거대한 그래프

2. 병렬 쿼리 시스템 : 스크립트난 사용자에게 친숙한 쿼리 인터페이스를 통해 병렬 처리할 수 있는 시스템 개발됐다.

가. 구글 Sawzall : Sawzall은 MapReduce 추상화한 스크립트 형태의 병렬 프로그래밍 언어다. Sawzall은 MapReduce를 추상화한 최초의 병렬 쿼리 언어이고, 이후에 나온 오픈소스 프로젝트인 Pig나 하이브도 개발 배경과 기본 개념은 Sawzall과 유사하다.

나. 아파치(야후) Pig : Pig는 야후에서 개발해 오픈소스 프로젝트화한 데이터 처리를 위한 고차원 언어다. Hadoop MapReduce 위에서 동작하는 추상화된 병렬 처리 언어이며 현재 아파치 하둡의 서브 프로젝트다.
- 개발 동기 : MapReduce는 Map과 Reduce 두 단계로 이루어진 단순 병렬 모델이다. 실제대부분의 업무는 한번의 MapReduce 작업으로 끝나는 것이 아니다. Map의 Output이 또 다른 Map의 Input으로 들어가야 하고 Reduce의 Output이 다른 Map의 Input으로 들어가야하는 Chaining이 되어야 하고 MapReduce 자체적으로는 지원하기가 어려웠다.

- 사용 예제 및 현황 : MapReduce는 기본적으로 무공유 구조이므로 Join 연산이 매우 복잡하나 Pig로 처리하면 간단하게 해결할 수 있다. 야후 내부의 검색 인프라, 광고 연관성 분석, 사용자

의도 분석, 검색엔진 쿼리 분석, Hoffman's PLSI 등 다양하게 활용되고 있다.

다. 아파치 하이브 : 하이브는 페이스북에서 개발한 데이터 웨어하우징 인프라다. Pig와 마찬가지로 하둡 플랫폼 위에서 동작하며 사용자가 친숙한 SQL 기반의 쿼리 언어와 JDBC 지원한다. 하둡에서 가장 많이 사용되는 병렬 처리 기능인 Hadoop-Streaming을 쿼리 내부에 삽입해 사용할 수 있다.

- 개발 배경 : 페이스북은 상용 DBMS 기반의 데이터 웨어하우스 시스템을 운영하고 있었다.시간이 지나며 데이터 규모가 늘어 관리, 운영 비용의 절감 필요성이 대두되었다. 이에 따라 상용 DBMS에서 하둡으로 교체하였고, 교체 과정에서 필요한 기능들, 사용자를 위한 커맨드 라인 인터페이스(CLI), 코딩 없이 애드푹(Ad-hoc) 질의할 수 있는 기능, 스키마 정보들의 관리 기능들을 하나씩 구현하면서 지금의 하이브라는 시스템 만들어졌다.

- 하이브 아키텍처 : 하이브 구성요소 중 MetaStore는 Raw File들의 콘텐츠를 일종의 테이블의 칼럼처럼 구조화된 형태로 관리할 수 있게 해주는 스키마 저장소다. 별도의 DBMS 설정하지 않으면 Embedded Derby를 기본 데이터베이스로 사용한다. 앞 단에는 커맨드 라인 인터페이스(CLI)가 있는데 사용자는 이 CLI 통해 Join나 Group by 같은 SQL 쿼리를 한다. 그러면 파서(Parser)에서 쿼리를 받아 구문 분석을 하고 MetaStore에서 테이블과 파티션 정보 참조해 Execution Plan을 만들어 낸다. 이 Plan을 Execution Engine에 보낸다. Excution Engine은 하둡의 JobTracker, 네임노드와 통신을 담당하는 창구 역할을 하며 MapReduce 작업을 실행하고 파일을 관리한다. cf) SerDe(Serialize + Deserializer 줄임말, 테이블 로우나 칼럼의 구분자 등 저장 포맷을 정의해주는 컴포넌트다. 하둡의 InputFormat과 OutputFormat 해당한다고 볼 수 있다.

- 하이브 언어 모델
• DDL(Data Definition Language) : 테이블 생성(Create Table), 삭제(Drop Table), 변경(Rename Table) 명령/테이블 스키마 변경(Alter Table, Add Column)/테이블 조회(Show Table), 스키마 조회(Describe Table)
• DML(Data Manipulation Language) : 로컬에서 DFS로 데이터 업로드(Load Data)/쿼리 결과를 테이블이나 로컬 파일 시스템, DFS에 저장
• Query : Select, Group by, Sort by, Joins, Union, Sub Queries, Sampling, Transform

3. SQL on Hadoop : 하둡의 제약사항 극복하기 위한 실시간 SQL 질의 분석 기술로 하둡상에 저장된 대용량 데이터를 대화 형식의 SQL 질의를 통해 처리하고 분석한다.

가. 임팔라 개요 : 임팔라는 분석과 트랜잭션 처리를 모두 지원하는 것을 목표로 만들어진 SQL 질의 엔진이다. 하둡과 Hbase에 저장된 데이터 대상으로 SQL 질의할 수 있다. 고성능을 위해 C++언어 이용하였으며 맵리듀스를 사용하지 않고 실행 중에 최적화된 코드 생성해 데이터 처리한다.

나. 임팔라 동작 방식 : 모든 노드에 임팔라 데몬이 구동되며 사용자는 이 데몬들이 구동된 임의의 노드에 JDBC나 ODBC 또는 임팔라셸을 이용하여 질의를 요청할 수 있다. 사용자 질의는 데이터 지역성을 고려해 노드 전체로 분산되어 수행된다. 사용자의 질의 요청을 취합하여 결과 값을 만들어서 사용자에게 제공한다. 실제 운영 환경에서는 **라운드로빈 방식**으로 사용자 질의를 분산시켜 전 노드들이 질의에 대해 코디네이터 역할을 고르게 수행할 수 있도록 해야한다.

다. 임팔라 SQL 구문 : 하이브의 SQL 이용한다.

데이터 정의 언어 (Data Definition Language)	데이터베이스, 테이블 생성 Create Database/Table
	테이블 변경, 파티션 추가 : Alter Table
	데이터베이스, 테이블 삭제 : Drop Database/Table
	데이터베이스, 테이블 조회 : Show Database/Table, Describe Database
데이터 조작 언어 (Data Manipulation Language)	데이터 조회 : Select, Where, GroupBy, OrderBy 구문 지원
	데이터 입력 : Insert into/overwrite 구문 지원
	데이터 변경 구문은 지원 안함
	데이터 삭제(Delete) 구문은 지원 안하나 테이블 삭제(Drop)시 데이터 삭제됨
내장 함수 (Builtin Functions)	수학함수 : 절대값(abs), 코사인값(acos), 로그값(log) 반환 등의 기능 제공
	타입 변환 : 날짜값(day) 반환, 유닉스에 포타임 변환(from_unixtime), 현재 시간(now) 반환 등 다수의 함수 제공
	조건문 : if문 제공, case 등 분기 기능 제공
	문자열 함수 : 아스키 코드값(asci) 변한, 문자열(concat) 병합, 정규 표현식(regexp)

라. 임팔라의 데이터 모델 : 임팔라는 하둡 분산 파일 시스템에 데이터를 저장한다. 어떤 저장 포맷을 사용하느냐에 따라 데이터 처리 성능이 달라진다. 하둡의 기본 파일 포맷인 텍스트나 시퀀스 파일은 로우 단위의 데이터 저장 방식을 사용한다.(로우 단위로 저장 시 테이블에서 하나의 컬럼을 읽든 전체 테이블을 읽든 동일한 디스크 입출력 발생) 컬럼 단위의 파일 저장 포맷인 RCFile 사용할 경우 데이터 처리 과정에서 발생하는 디스크 입출력 양 현저히 감소한다.(읽고자 하는 컬럼만큼의 디스크 입출력 발생하기 때문에 처리 성능 개선) 다만 하둡에 저장된 파일이 처음부터 컬럼 파일 포맷을 사용하지 않았을 경우 파일 포맷 변경 작업을 해주어야 한다.

제3절 클라우드 인프라 기술 : 클라우드 컴퓨팅은 동적으로 확장할 수 있는 가상화 자원들을 인터넷으로 서비스할 수 있는 기술을 말한다. 클라우드 서비스들은 SaaS(Software as a Service), PaaS(Platform as a Service), IaaS(Infrastructure as a Service) 3가지 유형으로 나뉜다. 클라우드 컴퓨팅에서 가장 근간이 되는 인프라 기술 중 서버 가상화 기술은 물리적인 서버와 운영체제 사이에 적절한 계층을 추가해 서버 사용하는 사용자에게 물리적인 자원 숨기고 논리적인 자원만 보여주는 기술 말한다. 서버 가상화는 하나의 서버에서 여러 개의 어플리케이션, 미들웨어, 운영체제들 서로 영향 미치지 않으며 동시에 사용할 수 있도록 해준다.
- 서버 가상화 기술 이용하여 얻을 수 있는 효과
• 가상머신 사이의 데이터 보호 : 가상머신 사이에는 보안적으로 분리되어 데이터 보호
• 예측하지 못한 장애로부터 보호 : 가상머신에서 수행 중인 애플리케이션의 장애가 다른 가상머신에는 전혀 영향 미치지 않는다.
• 공유 자원에 대한 강제 사용의 거부 : 하나의 가상머신은 할당된 자원 이상 가져가는 것 차단할 수 있다. 이 기능을 통해 다른 가상머신에 할당된 자원의 부족 현상 차단할 수 있다.

• 서버 통합 : 기존 서버의 용량을 증설하고 가상머신을 추가하여 동일한 데이터센서의 물리적 자원(공간, 전원 등)을 이용하면서 더 많은 서버 운영 가능하다.
• 자원 할당에 대한 증가된 유연성 : 수시로 변화하는 각 가상머신의 자원 요구량에 맞춰 전체 시스템 자원을 재배치함으로써 자원 활용도를 극대화할 수 있다.
• 테스팅 : 새로운 서버를 추가하지 않아도 테스트 환경 구성 가능하다. 부하 테스트가 필요한 경우에도 일시적으로 자원을 줄이는 방법으로 부하 상황을 만들 수 있으며 다수의 부하 생성 역할을 수행하는 노드도 쉽게 추가할 수 있다.
• 정확하고 안전한 서버 사이징 : 필요한 자원만큼만 가상머신 할당할 수 있다.
• 시스템 관리 : 마이그레이션 기능을 이용할 경우 운영 중인 가상머신의 중지없이 가상머신을 다른 물리적 서버로 이동시킬 수 있다. 하드웨어 장애, 로드밸런싱(특정 가상 서버나 가상 서버가 수행 중인 물리적 서버에 부하 집중되는 경우), 업그레이드 시 쉽게 수행 가능

가. CPU 가상화 : 하이퍼바이저(Hypervisor)는 물리적 서버 위에 존재하는 가상화 레이어 통해 운영체제가 수행하는데 필요한 하드웨어 환경을 가상으로 만들어준다. 일반적으로 가상머신을 하이퍼바이저라고 할 수 있다. 하이퍼바이저가 서버 가상화 기술의 핵심으로 x86계열 서버 가상화에서는 소프트웨어 기반으로 하이퍼바이저 구성한다. 하이퍼바이저는 VMM(Virtual Machine Monitor)라고도 하며 하드웨어 환경 에뮬레이션(Emulates a complete hard environment), 실행환경 격리(Isolate execution in each VM), 시스템 자원 할당(Allocates platform resources-processing, memory, I/O, storage), 소프트웨어 스택 보존(Encapsulates software stacks including the OS and state information)과 같은 기능 수행한다.

Container-based Virtualization (OpenVZ, Linux-Vserver, FreeBSD Jails)	
Hosted Virtualization 호스트 운영체제와 게스트 운영체제 사이 위치	
Bare-metal Virtualization 하드웨어와 호스트 운영체제 사이 위치	Full Virtualization 완전가상화 (VMware ESX, MS Virtual PC/Server)
	Para Virtualization 반가상화 (Denali, Xen)

• x86 계열 운영체제는 자신의 모든 하드웨어에 대한 제어 소유권을 갖고 있다는 가정 아래 하드웨어에 직접 명령을 수행하는 방식으로 디자인되어 있다. x86 아키텍처는 하드웨어에 대한 접근권한 관리를 위해 Ring0, 1, 2, 3 등 4개 레벨로 구성돼있다. 사용자 애플리케이션은 Ring3 레벨로 수행되며 운영체제의 경우 메모리나 하드웨어에 직접 접근해야하기 때문에 Ring0레벨에서 수행된다.
- 완전가상화 : 모든 자원을 하이퍼바이저가 직접 제어, 관리하기 때문에 어떤 운영체제라도 수정하지 않고 설치 가능한 장점이 있다. 하지만 하이퍼바이저가 자원 직접 제어하기 때문에 성능에 영향을 미친다. 자원들이 하이퍼바이저에 밀접하게 연관되어 있어 운영 중인 게스트 운영체제에 할당된 CPU나 메모리 등의 자원에 대한 동적 변경 작업이 단일 서버 내에서는 어렵다.(VMware의 VMotion과 같은 솔루션 도움 필요) 완전가상화는 하이퍼바이저보다 우선순위가 낮은 가상머신에서는 실행되지 않는 privileged 명령어에 대해서 trap을 발생시켜 하이퍼바이저에서 실행하는 방식으로 MS윈도우와 같은 Guest OS가 하이퍼바이저 상에서 변경되지 않은 상태로 실행될 수 있는 장점이 있으나 반가상화에 비해 속도가 느리다.
- 하드웨어 지원 완전가상화 : 가상머신에서 메모리와 CPU 등의 하드웨어에 명령을 내릴 수 있는 반가상화 수준의 성능을 발휘하도록 개선하고 있다. 하이퍼바이저는 Ring-1에서 수행되고 가상머신의 운영체제(Guest OS)는 Ring0에서 수행되어 privileged 명령어에 대해 추가로 변환 과정이 필요없다. 하이퍼바이저를 거쳐 바로 하드웨어로 명령이 전달되어 성능↑ • 하드웨어 지원 가상화 사용할 경우 CPU 사용률 높아진다. 특히 I/O나 메모리를 많이 사용하는 경우 CPU 사용률이 높아진다. 따라서 서버 통합 목적일 경우 비효율적이다.(인텔에서 하드웨어 지원 가상화 사용시 주의사항 발표)

- 반가상화 : privileged 명령어를 게스트 운영체제에서 hypercall로 하이퍼바이저에 전달하고 하이퍼바이저는 hypercall에 대해서는 privilege 레벨에 상관없이 하드웨어로 명령을 수행시킨다. Hypercall은 게스트 운영체제에서 요청을 하면 하이퍼바이저에서 바로 하드웨어 명령을 실행하는 call을 말한다. 반가상화 기반에서는 CPU와 메모리 자원의 동적 변경이 서비스의 중단 없이 이루어질 수 있으며, 완전가상화에 비해 성능이 뛰어나다. 반가상화는 privileged 명령어를 직접 호출(hypercall)하므로 속도는 빠르나 커널을 변경해야 하고 완전가상화는 dynamic binary translation(Xen은 emulation) 모듈과의 통신을 통해 처리하므로 속도는 느리나 커널 변경은 없다.

- Monolithic vs Microkernel : 하드웨어에 대한 드라이버가 어느 계층에 있느냐에 따라 Monolithic 방식(가상머신이 I/O를 위해 하드웨어에 접근할 때 사용하는 드라이버를 하이퍼바이저 계층에서 모두 갖고 있는 방식)과 Microkernel 방식(각 가상머신에서 드라이버를 갖는 방식)으로 구분한다. ex) VMware : Monolithic(하이퍼바이저가 드라이버 가지고 있으며 모든 I/O 요청은 하이퍼바이저가 수행), Xen : Microkernel(하이퍼바이저는 드라이버 없어 호스트 운영체제가 드라이버를 가지고 있고 각 게스트 운영체제는 가상 드라이버가지고 있어 I/O 요청을 위해 호스트 운영체제를 거쳐야 한다. 게스트와 호스트 운영체제는 서로 격리되어 있어 하이퍼바이저(또는 VMBus)를 이용해 요청 주고 받는다.)

• Monolithic 방식은 성능은 조금 향상 가능하지만 하이퍼바이저에서 모든 드라이버를 가지고 있어야 하기 때문에 하드웨어 추가되거나 드라이버 업뎃되는 경우 하이퍼바이저가 수정되어야 하고 더 많은 코드를 가지고 있기 때문에 장애 발생 가능성도 높다. Microkernel 방식의 경우 속도는 조금 느리지만 하이퍼바이저 계층이 간단하여 드라이버 업뎃이나 하드웨어 추가에 따른 하이퍼바이저 변경이 필요 없으며 장애 발생 확률도 낮다.

구분	완전가상화 (CPU기술 이용X)	완전가상화 (CPU기술 이용)	반가상화
사용기술	바이너리 변환, Direct Execution	Privileged Instruction은 Ring-1로 처리됨	수정된 OS 사용
게스트 OS 변경/호환성	게스트 OS 변경 없음, 호환성 뛰어남	게스트 OS 변경 필요 없음, 호환성 뛰어남	Hypercall 가능하도록 게스트 OS 변경함,

		(단, CPU 지원해야함)	호환성 안좋음
성능	좋음	Fair(점점 바이너리 변환 방식 성능에 근접)	특정 경우에 더 좋음
제품	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen
게스트 OS가 하이퍼바이저에 독립적인지	독립적	독립적	Xen Para Virtualization은 Xen 하이퍼바이저에서만 동작, VMI 규격 따르는 VMI-Linux는하이퍼바이저에 독립적

- 호스트 기반 가상화 : 완전한 운영체제가 설치되고 가상화를 담당하는 하이퍼바이저가 호스트 운영체제 위에 탑재되는 방식이다. 가장 큰 단점은 단일 운영체제의 취약성에 있다. ex) VMware Workstation, Microsoft Virtual PC

- 컨테이너 기반 가상화 : 호스트 운영체제 위에 가상의 운영체제를 구성하기 위한 운영 환경 계층을 추가하여 운영체제만을 가상화한 방식이다. 운영체제만을 가상화 대상으로 하므로 전체 하드웨어를 대상으로 하는 하이퍼바이저 기반 가상화 방식에 비해 훨씬 적게 가상화한다. 컨테이너 기반 가상화 방식에서 가상화를 지원하는 계층을 하이퍼바이저라고 하지 않고 가상 운영환경(Virtual server environment)라고 부른다. 가상화 수준이 낮기 때문에 다른 방식에 비해 빠른 성능 보이지만 자원 간 격리 수준이 낮아 하나의 가상 운영체제에서 실행되는 애플리케이션의 자원 사용에 따라 다른 가상 운영체제가 영향 받는 단점이 있다. 또한 호스트 운영체제의 보안 취약하다.

구분	하이퍼바이저 기반(Full, Para)	컨테이너 기반
하드웨어 독립성	가상머신 내에서 완전 독립	호스트 OS 사용
OS 독립성	호스트 OS와 완전 독립 (리눅스와 윈도우 머신 동시 사용)	호스트와 게스트 동일
격리 수준	높음	낮음
성능	높은 오버헤드 발생 성능 향상 위해 HW 가상화 기술 병행	오버헤드 거의 없음 HW 자원 대부분 호라용
관리	가상머신 별로 별도 관리	공통 SW 중앙 집중식 관리
응용 분야	이기종 통합 (윈도우와 리눅스 혼합 환경)	단일 OS 환경 자원 통합, 대규모 호스팅 업체
대표 제품	VMware ESX, MS Virtual Server XEN(Para Virtualization)	Virtuozzo(상용, OpenVZ-공개) Sun Solaris Container

나. 메모리 가상화 : (VMware의 기법)운영체제는 메모리를 관리하기 위해 물리주소(Physical Address)와 가상주소(Virtual Address) 두가지를 사용하고 있다. 물리주소는 0부터 시작해서 실제 물리적인 메모리 크기까지 나타내고 가상주소는 하나의 프로세스가 가리킬 수 있는 최대 크기를 의미하며 32비트 운영체제에서 4GB까지 가능하다. 프로그램에서의 주소는 물리적인 메모리의 주소 값이 아닌 가상주소 값이다. 따라서 가상주소 값의 위치(VPN, Virtual Page Number)를 실제 물리적 주소 값 위치(MPN, Machine Page Number)로 매핑 과정이 필요하며 page table을 이용한다. 매핑 연산을 하드웨어적으로 도와주는 것을 TLB(Translation Lookaside Buffer)라고 한다. VMware의 하이퍼바이저 핵심 모듈을 VMKernel라고 한다. VMKernel은 Service Console, 디바이스 드라이버들의 메모리 영역을 제외한 나머지 전체 메모리 영역을 모두 관리하면서 가상머신에 메모리를 할당한다. 생성되는 가상머신은 자신에게 할당된 메모리들을 연속된 공간의 실제 물리적인 메모리로 인식한다.

• VMware는 하이퍼바이저 내 Shadow Page Table을 별도로 두어 가상 메모리 주소와 물리 메모리 주소의 중간 변환 과정 가로챈다. 이 테이블은 마치 연속된 빈 공간의 메모리가 실제 존재하는 것처럼 게스트 운영체제에게 매핑해주는 역할을 하며 동시에 개별적인 모든 가상머신들이 자신만의 메모리 주소 공간을 갖도록 한다.

- Memory ballooning : VMKernel은 예약된 메모리보다 더 많은 메모리를 사용하는 가상머신의 메모리 영역을 빈 값으로 강제로 채워 가상머신 운영체제가 자체적으로 swapping하도록 한다. 가상머신 운영체제에서 보이는 물리적인 메모리(실제 하이퍼바이저에서 제공한 논리적 메모리)가 채워지고 있는 것을 감지한 가상머신 운영체제는 swap 파일에 메모리 영역을 page out 시키고 메모리를 비운다. 하이퍼바어지는 page out된 메모리 영역을 다른 가상머신에 할당한다.

- Transparent page sharing : 하나의 물리적 머신에 여러 개의 가상머신 운영되는 경우 가상머신 할당된 메모리 중 동일한 내용 담고 있는 페이지는 물리적 메모리 영역에 하나만 존재시키고 모든 가상머신이 공유한다.

- Memory Overcommitment : 2GB 메모리 가진 물리적 장비에 512MB를 Minimum reserved를 가질 수 있는 가상머신 5개를 수행할 수 있지만 모든 가상머신이 메모리 사용이 많은 업무 수행하는 경우에는 성능저하 발생할 수 있으므로 권장하지 않는다.

다. I/O 가상화 : 하나의 물리적 장비에서 여러 가상머신 실행 시 가장 문제되는 것은 I/O에서의 병목현상이다. CPU 자원의 파티셔닝만으로는 가상화 기술을 제대로 활용할 수 없으며 I/O 자원의 공유 및 파티셔닝 필요하다. 또한 물리적 머신에서 운영되는 가상머신 간에도 통신이 이루어져야 한다. 이를 위해 가상 디스크 어댑터, 가상 이더넷 어댑터, 공유 이더넷 어댑터 등과 같은 기술들이 사용된다.

- 가상 이더넷 : 대표적인 I/O 가상화 기술의 하나로 가상화 기능 중 물리적으로 존재하지 않는 자원을 만들어내는 에뮬레이션 기능을 이용한다. 가상 이더넷을 이용할 경우 각 가상머신들 사이에 물리저인 네트워크 어댑터 없이도 메모리 버스를 통해 고속 및 고효율 통신 가능하다. 또한 가상 이더넷은 가상 LAN 기술 기반으로 네트워크 파티션 가능하게 한다. 가상 이더넷을 통해 사용자들은 별도의 물리적 어댑터와 케이블 사용하지 않고 네트워크의 이중화 및 안정적 단절 등의 효과 얻을 수 있다.

- 공유 이더넷 어댑터 : 여러 개의 가상 머신이 물리적인 네트워크 카드를 공유할 수 있게 하며, 공유된 물리적 카드 통해 외부 네트워크와 통신 가능하다. 하나의 자원을 이용하여 여러 가상머신이 공유하기 때문에 발생하는 병목현상은 피할 수 없다.

- 가상 디스크 어댑터 : 한대의 서버가 여러 개의 가상머신 구성할 경우 가장 문제점은 외장 디스크를 사용할 수 있게 해주는 파이버 채널 어댑터(Fiber Channel Adapter)와 같은 I/O 어댑터의 부족이다. 이를 위해 가상 디스크 어댑터가 필요하다.

• 가상화된 환경에서 가상 디스크를 이용해 가상머신이 디스크 자원 획득하는 방법

① 내장 디스크의 경우 가상 I/O 레이어가 내장 디스크들을 소유하고 있고 내장 디스크들을 논리적 디스크 드라이브로 나눈다. 논리적으로 나누어진 드라이버는 LUN(Logical Unit Number)으로 각 파티션에 가상 디스크 어댑터 통해 분배된다. 해당 가상머신은 이렇게 획득한 논리적 디스크 자원을 물리적 자원처럼 인식한다.

② 외장 디스크의 경우 가상 I/O 레이어가 파이버 채널 어댑터를 통해서 외장 디스크의 LUN을 획득한다. 가상 I/O 레이어가 이 자원을 바로 각 가상머신에 가상 디스크 어댑터 통해 분배한다. 이처럼 가상 I/O 레이어 통해 제공된 논리적 디스크 볼륨은 이를 이용하는 다른 가상머신에게는 SCSI 디스크로 나타난다.