고용노동부 한국산업인력공단 국가인적자원개발컨소시엄

빅데이터

# R을 이용한 데이터 마이닝

(교육기간 : 2018년 08월 25일 ~ 09월 01일)

지태창

대한민국 소프트웨어 기술진흥 · 인력양성 대표기관
한국소프트웨어기술진흥협회

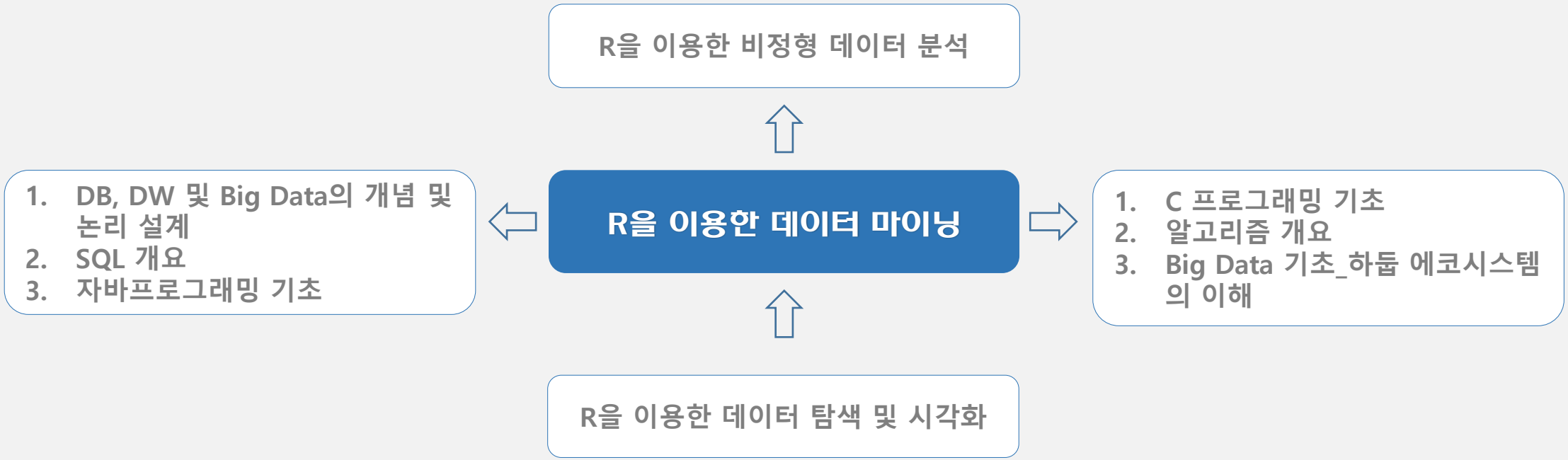
# 과정개요

- 데이터 마이닝에 대한 일반적인 이해와 더불어 OSS R을 이용한 데이터 마이닝 분석 방법에 대한 활용
- 데이터 마이닝에 대한 이론적인 이해 및 R을 이용한 실습
- 상황에 따른 데이터 마이닝 적용 방법론에 대한 이해 및 만들어진 데이터 마이닝 모델에 대한 평가

# 과정목표

- OSS R을 활용하여 분석할 수 있는 다양한 기본적인 분석 기법을 이해하고, 데이터 마이닝을 위한 방법론 및 기본적인 개념을 이해할 수 있습니다.
- 데이터 마이닝 모델의 학습, 평가에 대한 지식을 습득하고, Clustering, Regression, Neural Networks, Decision Tree 등에 대한 개념을 학습할 수 있습니다.

# 러닝 맵


R을 이용한 비정형 데이터 분석
1. DB, DW 및 Big Data의 개념 및 논리 설계
2. SQL 개요
3. 자바프로그래밍 기초
R을 이용한 데이터 마이닝
1. C 프로그래밍 기초
2. 알고리즘 개요
3. Big Data 기초_하둡 에코시스템의 이해
R을 이용한 데이터 탐색 및 시각화

# 강사 프로필

| | |
|---|---|
| 성명 | 지태창 |
| 소속 및 직함 | NH농협캐피탈 / 부장 |
| 주요 경력 | NH농협캐피탈 (2018 ~ 현재)<br>LG CNS (1999 ~ 2018) |
| 강의/관심 분야 | 데이터 분석, 인공지능 |
| 자격/저서/대외활동 | 컴퓨터과학 박사 |

# 학습모듈(Learniong Object) 및 목차

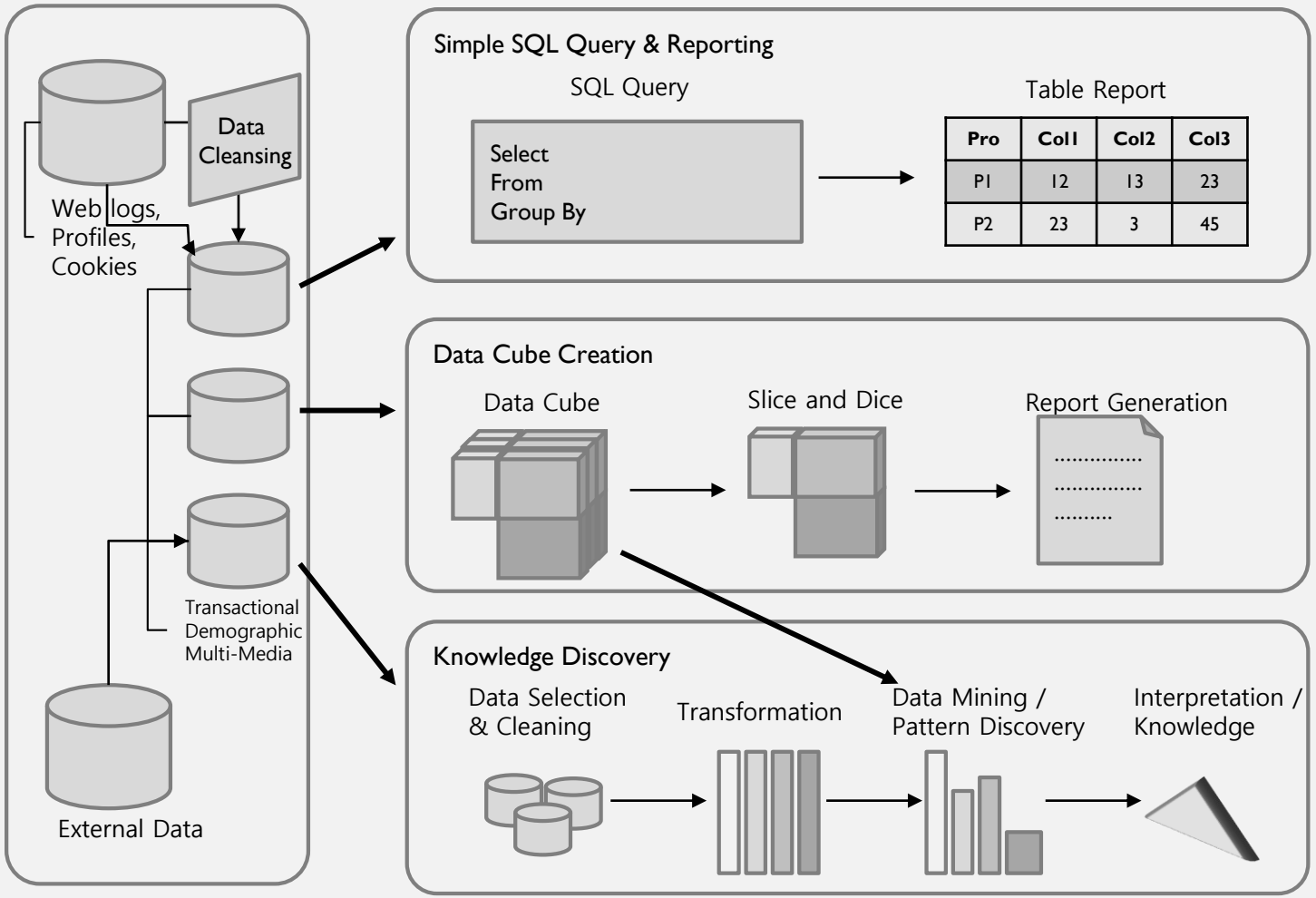
| LO명 | LO별 목차 |
| --- | --- |
| 데이터 마이닝과 R | 1. 데이터 마이닝 소개<br>2. R 개발 환경 소개<br>3. 데이터 소스 정의<br>4. 데이터 탐색 |
| 분류 (Classifications) | 1. 분류 개념 소개<br>2. Decision Trees<br>3. Neural Networks<br>4. SVM |
| 회귀 분석(Regression) | 1. 회귀 분석 개념 소개<br>2. Linear Regression |
| 군집화 (Clustering) | 1. 군집화 개념 소개<br>2. K-Means<br>3. Hierarchical Clustering |

## 학습지식 개요/요점

- 데이터로부터 숨겨진 지식을 발견하기 위한 데이터 마이닝에 대한 기본 개념을 이해할 수 있다.
- R의 기본 개발 환경을 이해할 수 있다.
- R에 대한 기본 개념을 이해할 수 있다.
- R을 이용한 데이터 처리 방법을 살펴보고, 직접 실행해 본다.

# Contents

# Computational Knowledge Discovery



# 용어

- 데이터 마이닝 (Data Mining)
  - Knowledge Discovery Process의 한 단계
  - 특별한 알고리즘들로 구성
  - 데이터에서 특정 패턴 / 모델을 생성

- 지식 탐색 과정 (Knowledge Discovery Process)
  - 데이터 마이닝 방법론을 사용하는 프로세스
  - Knowledge (지식)을 추출
  - 데이터베이스에 대한 다양한 전 처리와 변환 과정 수행

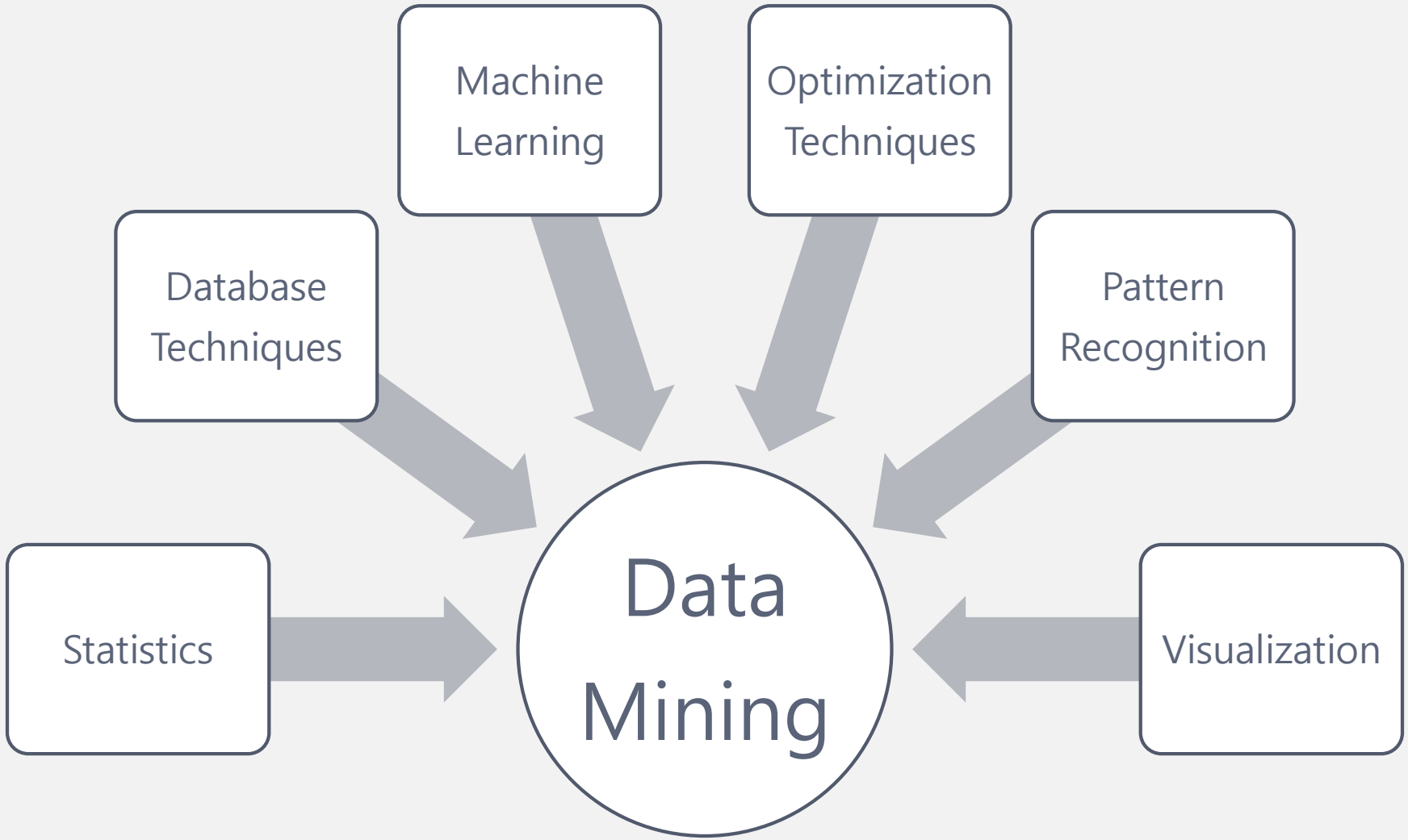
# 데이터 마이닝의 정의

- 대규모 데이터베이스에 숨겨진 예측 정보를 자동으로 추출하는 과정
- Three key words:
  - 자동 (Automated)
  - 숨겨진 (Hidden)
  - 정보 (Predictive Informaction)
- 데이터 마이닝은 상황을 주도하게 함
  - Retrospective (회고)가 아닌 Prospective (미래의)

# 데이터 마이닝의 목적

- 데이터 소스에서 모델 생성까지의 전반적인 통계적인 과정을 단순화하고 자동화하는 것
- 다양한 데이터 마이닝 알고리즘과 도구 존재
- 다른 기법들의 결과를 비교하기 위한 통계적인 전문 지식 필요
- **소프트웨어에 지능을 부여하는 것**

# 연관 분야


Machine Learning
Optimization Techniques
Database Techniques
Pattern Recognition
Statistics
Data Mining
Visualization

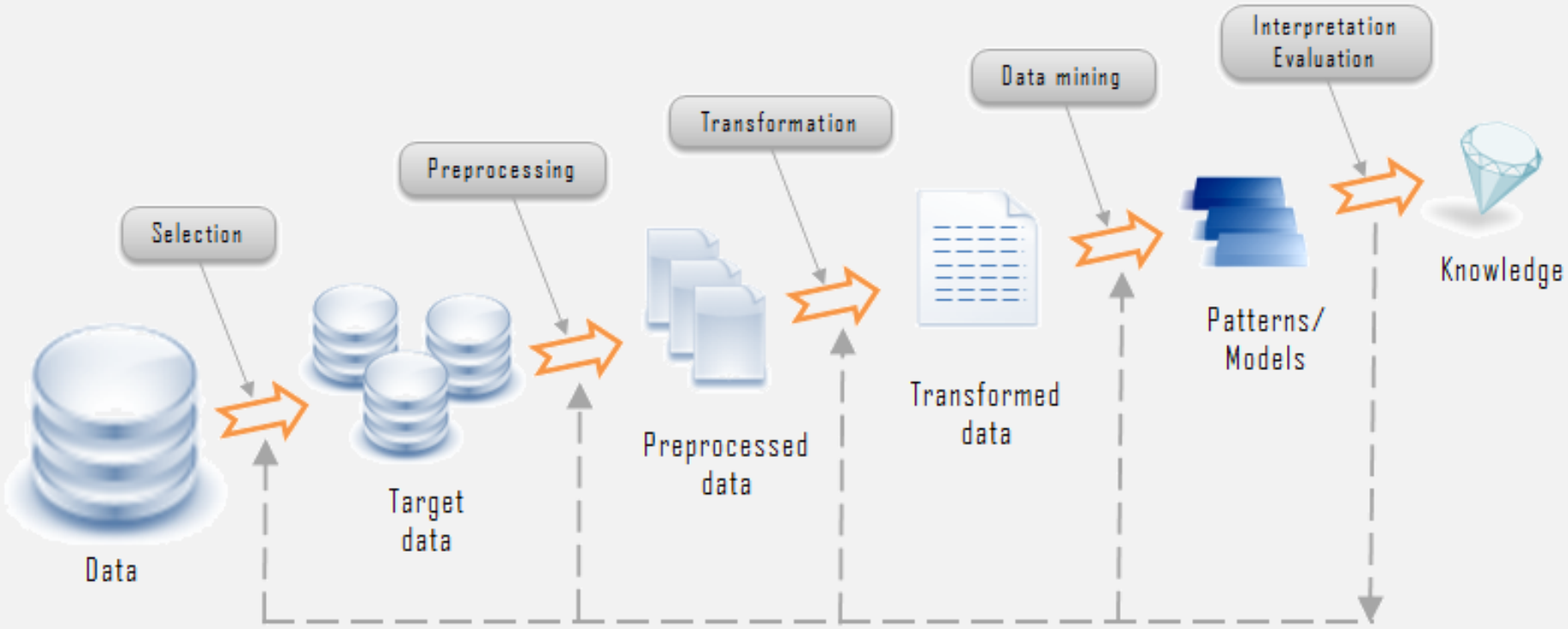
# 필요성

- 전통적인 분석 방법론을 사용하기에는 데이터가 너무 크다.
  - 데이터 개수의 증가 ($10^9$ – $10^{12}$ Bytes -> GB - TB)
  - 다차원 데이터 (100 - 1000 속성)
- 기업의 데이터 자산
  - 소규모(5-10 %) 데이터 만 분석
  - 분석할 수 없는 데이터도 저장
- 데이터베이스가 확대되면서 전통적인 query language에 의한 의사 결정이 어려워 짐
  - Query language에 의해서 다양한 관점을 분석하기 어려움

# 데이터 마이닝은

- 궁극적으로 데이터에서 이해할 수 있는 패턴을 찾아내는 것
- 이해할 수 있는 패턴
  - 새로운 데이터에 대한 예측과 분류 수행
  - 기존 데이터에 대한 설명
  - 의사 결정을 지원하기 위한 대규모 데이터베이스에 대한 요약
  - 심도 깊은 패턴을 발견하기 위해 사람을 도와줄 수 있는 Graphical Data Visualization

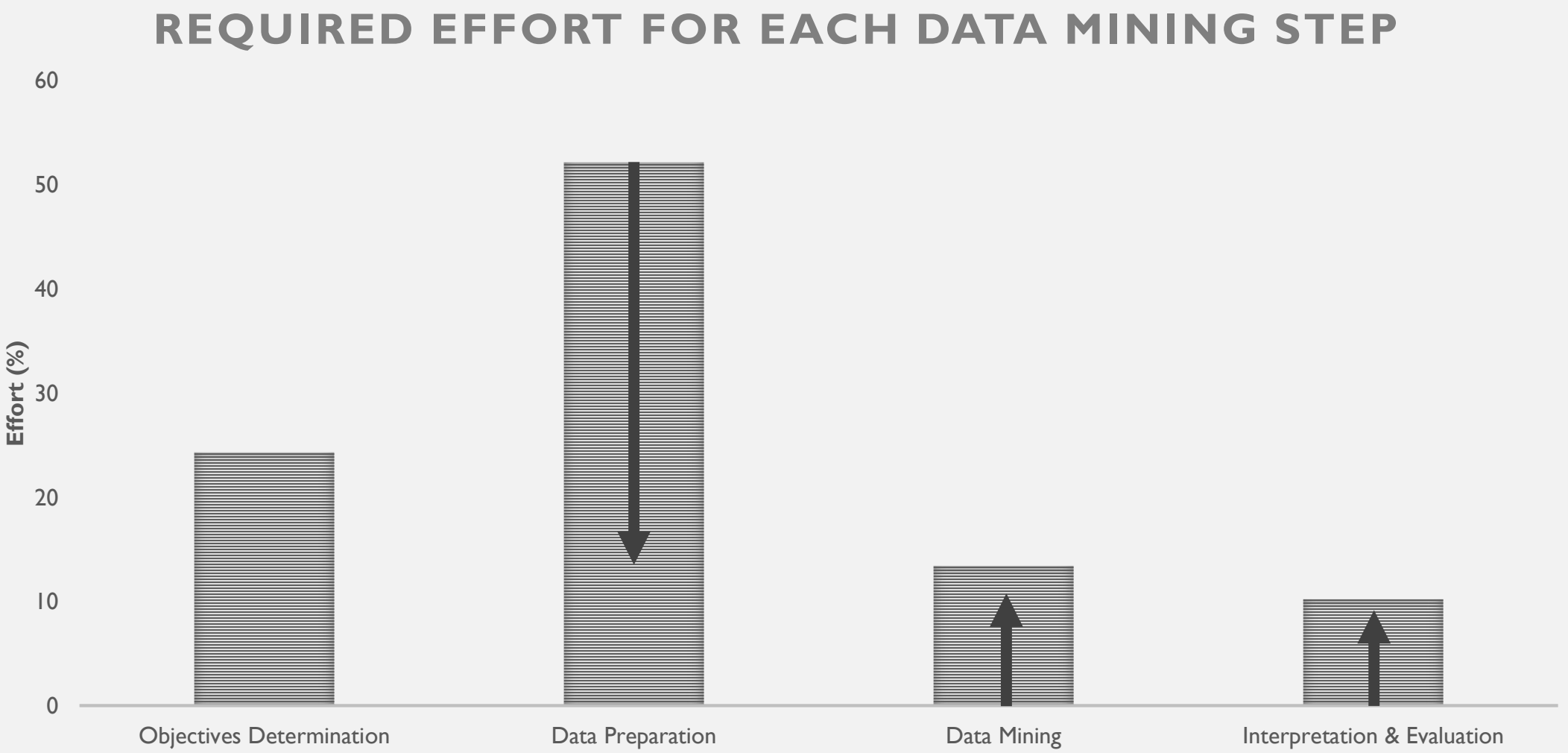
# 데이터 마이닝 수행 절차


Selection
Preprocessing
Transformation
Data mining
Interpretation
Evaluation
Data
Target
data
Preprocessed
data
Transformed
data
Patterns/
Models
Knowledge

# 데이터 마이닝 수행 절차

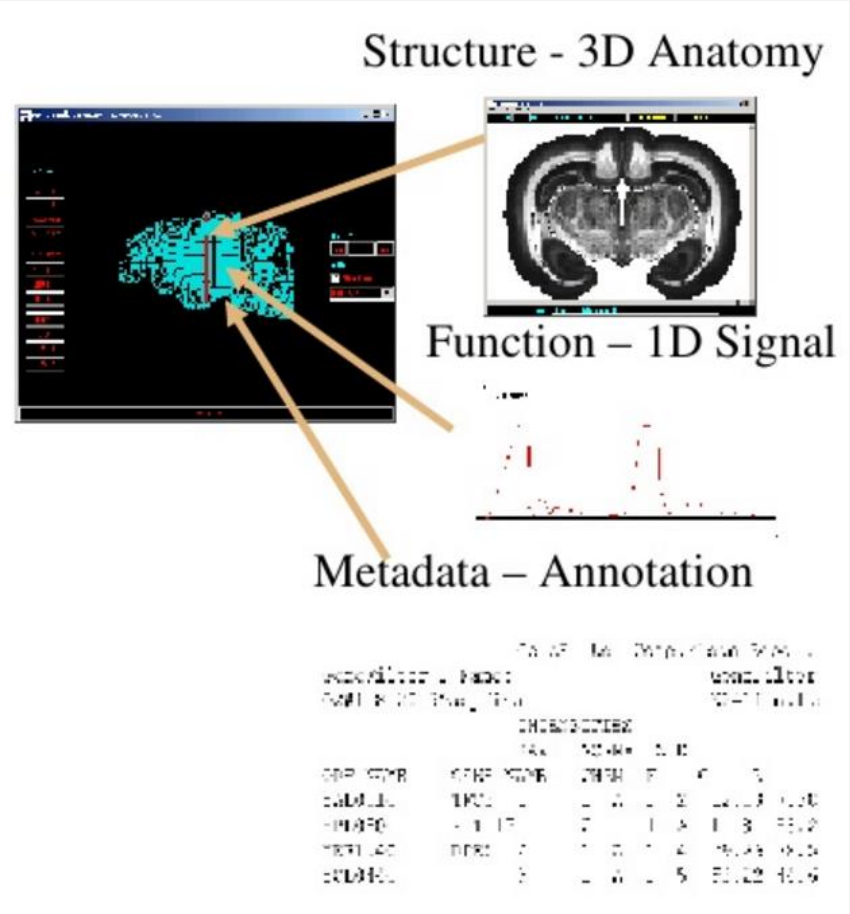
- 데이터 수집 (Data Selection)
  - 텍스트파일, 엑셀파일, DB 테이블 등 다양한 형태로 저장 및 수집이 가능
- 데이터 탐색 및 전처리 (Data Preprocessing / Transformation)
  - 수집된 데이터를 분석 목적에 맞게 가공 및 데이터의 분포를 확인하여 모델 개발에 필요한 인사이트 추출
- 모델 생성 (Data Mining)
  - 분석 목적에 맞는 데이터 마이닝 모델을 선택하여 학습 데이터 셋에 모델 생성 수행
- 모델 평가 (Interpretation / Evaluation)
  - 테스트 데이터 집합에 생성된 모델을 적용시킨 후 사전 정의된 평가 지표에 의거하여 모델의 성능을 평가
- 모델 개선 (Data Mining)
  - 평가 결과가 요구 수준에 미치지 못하는 경우, 이전 단계를 반복하며 개선 활동 수행

# 데이터 마이닝 과정에 필요한 시간



# 데이터의 종류

- Relational Databases
- Data Warehouses
- Transactional Databases
- Advanced Database Systems
  - Object-Relational
  - Spatial and Temporal
  - Time-Series
  - Multimedia
  - Text
  - Heterogeneous
  - WWW



# 기계 학습(Machine Learning)

- 컴퓨터가 일일이 코드로 명시하지 않은 동작을 데이터로부터 학습하여 실행할 수 있도록 하는 알고리즘
- 주어진 자료들을 학습하여 일반화된 규칙 또는 새로운 지식을 자동으로 추출해내는 방법
- 학습을 통해 인간의 개입 없이 자동으로 주어진 업무를 처리할 수 있는 방법
- 최근 각광을 받고 있는 AI (Artificial Intelligence)의 한 분야

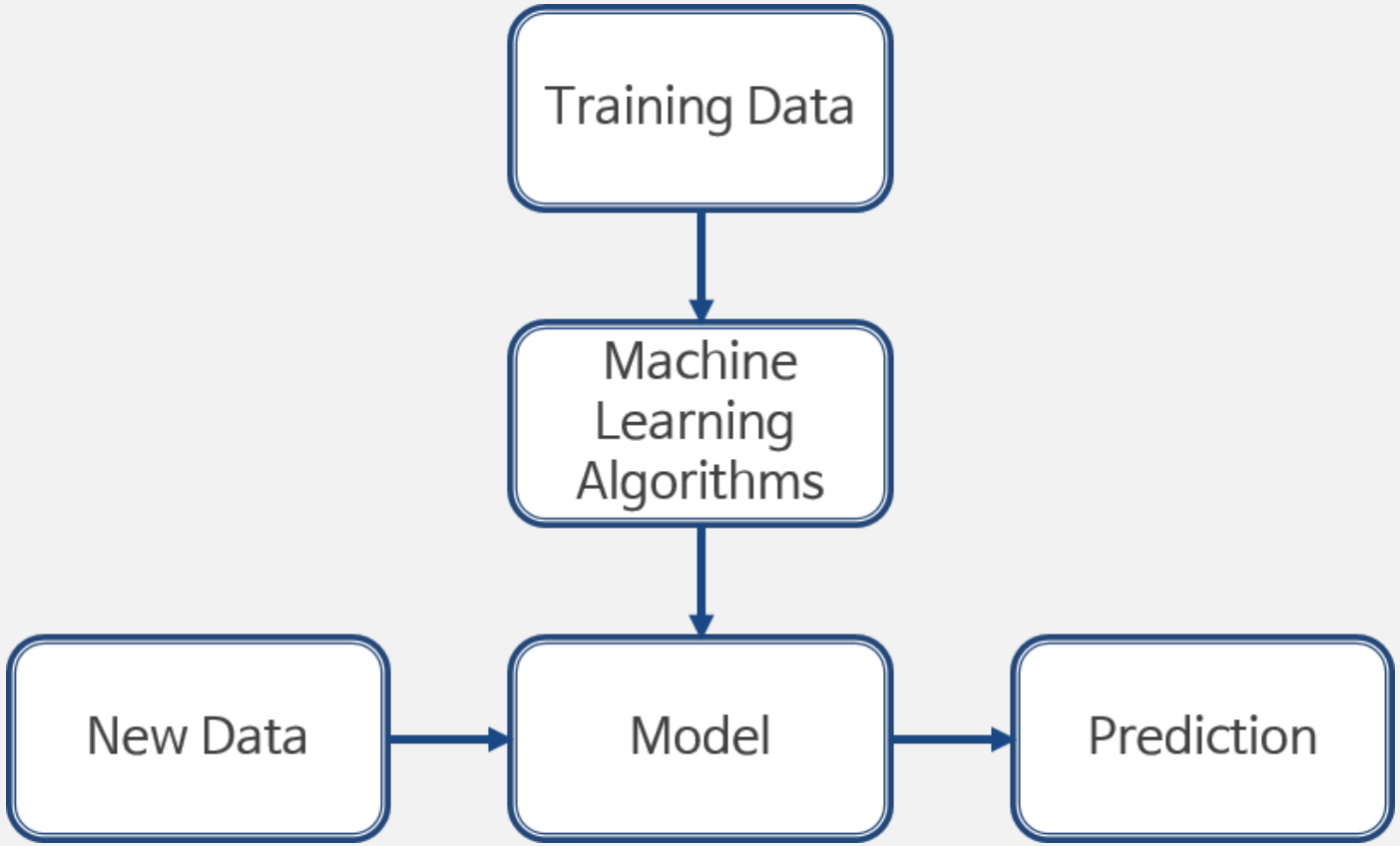
# 학습(Learnining)의 의미

- Tom M. Mitchell, Machine Learning, (1998)
- "만약 작업 T 에 대해 기준 P 로 측정한 성능이 경험 E 로 인해 향상되었다면, 그 프로그램은 작업 T 에 대해 기준 P 의 관점에서 경험 E 로부터 "배웠다"라고 말할 수 있다."

# 기계 학습을 쓰는 이유는?

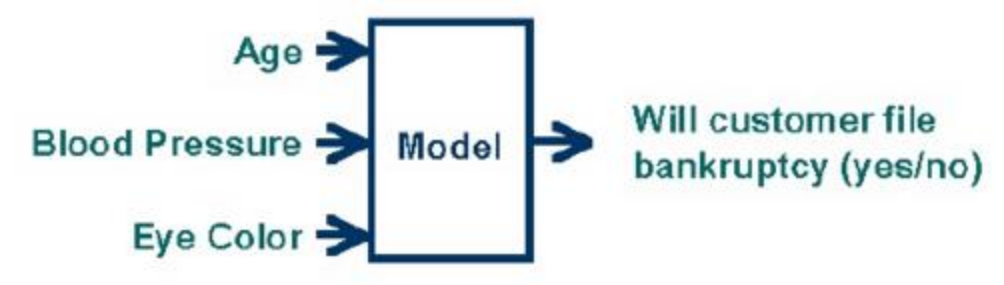
- 모든 문제를 확정적으로(Deterministically) 해결 할 수 없기 때문임
- 일반적인 프로그래밍으로 해결할 수 있는 케이스
  - 자동판매기 - 동전 투입, 음료수 선택, 거스름돈 계산 등 모든 작업들을 모두 사전에 정의하여 프로그램으로 만들 수 있음
- 기계 학습이 필요한 케이스
  - 스팸 이메일 필터기 - 스팸 이메일은 각 이메일마다 모두 다른 문구와 스타일을 가지고 있기 때문에 모든 스팸 이메일을 사전에 정의할 수 없음

# 기계 학습의 진행 프로세스 예시



# Predictive Model (예측 모델)

- 과거와 현재의 정보를 기반으로 미래를 예측하는 'black box'




- 다량의 입력이 필요함

# 스코어링 / 예측 (Scoring / Prediction)

- 모델은 한 번 만들어지고, 지속적으로 사용됨
- 데이터 마이닝 결과를 사용하는 사람은 데이터 마이닝 모델을 만드는 사람과는 다름
  - 모델을 전달하는 방법에 대한 고민 필요
- Issue: 모델 개발 시 사용된 데이터와 모델을 사용할 때 사용되는 데이터의 조화
  - 데이터가 동일한가?
  - 일관성이 자동적으로 유지되는가?

# 데이터 마이닝 응용 분야

- 시장 분석 (Market analysis)
- 위험 분석과 관리 (Risk analysis and management)
- 사기 / 이상치 탐색 (Fraud detection and detection of unusual patterns (outliers))
- 텍스트 마이닝 (Text mining (news group, email, documents) and Web mining)
- 스트림 데이터 마이닝 (Stream data mining)
- 바이오 인포메틱스 (DNA and bio-data analysis)

# 시장 분석 (Market Analysis)

- 데이터 소스
  - 신용 카드 거래 현황, 회원 카드, 할인 쿠폰, 고객 컴플레인
- 표적 마케팅 (Target marketing)
  - 동일 특성을 가지는 고객들의 군집 모델 생성 – 관심, 수입, 소비 형태 등
  - 지속적으로 고객의 구매 패턴 분석
- 교차 분석 (Cross-market analysis)
  - 제품 판매 사이의 관계 분석을 통한 예측 수행
- 고객 프로파일링 (Customer profiling)
  - 어떤 성향의 고객이 어떤 제품을 구매할 지 분석
- 고객 욕구 분석 (Customer requirement analysis)
  - 다양한 고객 사이의 최적의 제품 선정
  - 새로운 고객에 영향을 끼치는 요소 예측

# 회사 분석, 위험 분석

- 금융 계획, 자산평가 (Finance planning and asset evaluation)
  - 현금 흐름 분석과 예측
  - 시계열 분석 (재무 비율, 추세 분석 등)
- 자원 계획 (Resource planning)
  - 자원과 지출 비교 및 요약
- 경쟁사 분석 (Competition Analysis)
  - 경쟁사와 시장 방향 모니터링
  - 고객 군집화 및 군집 기반 가격 정책 수립
  - Red ocean 시장에서의 가격 정책 수립

# 사기 검출과 이상치 탐색

- 적용 영역 : 의료, 소매, 신용 카드, 통신 등
  - 자금 세탁 : 의심되는 통화 흐름 추적
  - 의료 보험
    - 전문적인 환자, 의사와의 유착 관계
    - 불필요하거나 유사한 진단
  - 통신 : 통화 사기
    - 통화 모델 : 통화의 목적지, 지속 시간, 요일의 시간. 예상 된 표준에서 벗어나는 패턴 분석
  - 소매업
    - 분석에 따르면 소매 손실의 38 %는 부정직 한 직원에 의한 것
  - 테러 방지

# 독립 변수 , 종속 변수

- 종속변수(Target variable)
  - 우리가 알고자 하는 결과값 (Target)
- 독립변수(Independent variable)
  - 종속변수(결과값)에 영향을 주는 입력 값 (Input)
- (예시) 아파트 시세 예측
  - 종속변수
    - 아파트 시세
  - 독립변수
    - 면적, 아파트 브랜드 가치, 역세권 여부 등등

# 데이터 마이닝 문제의 종류

- 분류 (Classification / Segmentation)
  - 이진 분류 (Binary : Yes/No)
  - 다중 분류 (Multiple category : Large/Medium/Small)
- 회귀 / 예측 (Regression / Forecasting)
- 연관 규칙 탐색 (Association rule extraction)
- 연속적인 흐름 탐색 (Sequence detection)
  - 휘발유 구매 → 보석 구매 : 사기
- 군집화 (Clustering)

# 지도 학습, 비지도 학습

- 지도 학습 (Supervised Learning) : 문제 해결, 교사 학습
  - 문제와 연결된 결과가 명확할 경우
  - 결과의 품질은 데이터의 품질에 관련
- 비지도 학습 (Unsupervised Learning) : 탐색 (군집화), 비교사 학습
  - 데이터에 대한 초기 이해에 유용
  - 불분명한 패턴이 나타나기도 함

# 지도학습이란?

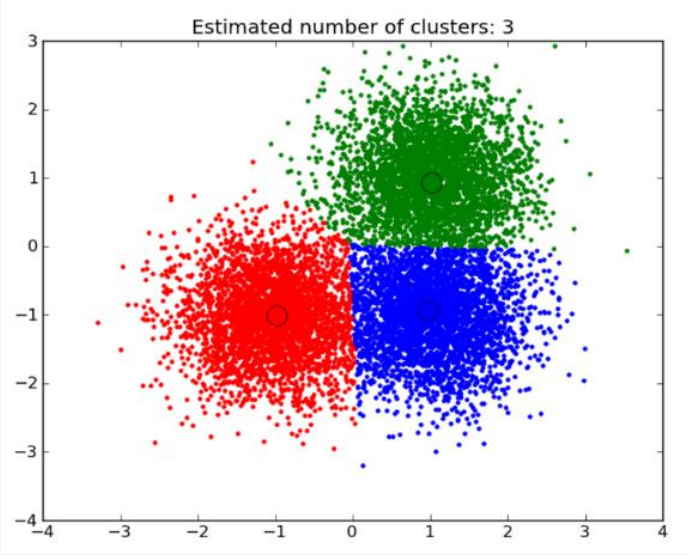
- 학습을 할 때 입력 값 (독립변수)과 입력 값에 해당하는 정답 (종속변수)를 함께 제공하여 모델을 생성
- 분류(Classification) 모델
  - 주어진 데이터가 속하는 범주 예측
  - 상품 추천, 스팸 이메일 필터링, 보험 사기 적발 등이 대표적 예
- 회귀(Regression) 모델
  - 주어진 데이터에 해당하는 수치(연속형) 예측
  - 환율 예측, 아파트 시세 예측 등이 대표적 예

# 분류(Classification) 모델

- 주어진 데이터가 속하는 범주 예측할 때 종속변수가 범주형일 경우, 범주를 예측할 수 있는 모델
- 종속변수의 범주는 2개 이상이며, 종속변수의 범수가 2개인 경우 이항 분류(Binary Classification), 3개 이상일 경우 다중클래스 분류(Multi-Class Classification) 이라 함
- 이항분류 (Binary Classification) 예시
  - 이메일의 스팸/ 비 스팸 이메일 여부 분류
    - 범주1: 스팸
    - 범주2: 비 스팸
- 다중 클래스 분류 (Multi-Class Classification) 예시
  - 영화 사이트 회원의 선호하는 영화 장르 추천
    - 범주1: 공포 영화
    - 범주2: 로맨틱 영화
    - 범주3: SciFi

# 비지도 학습이란?

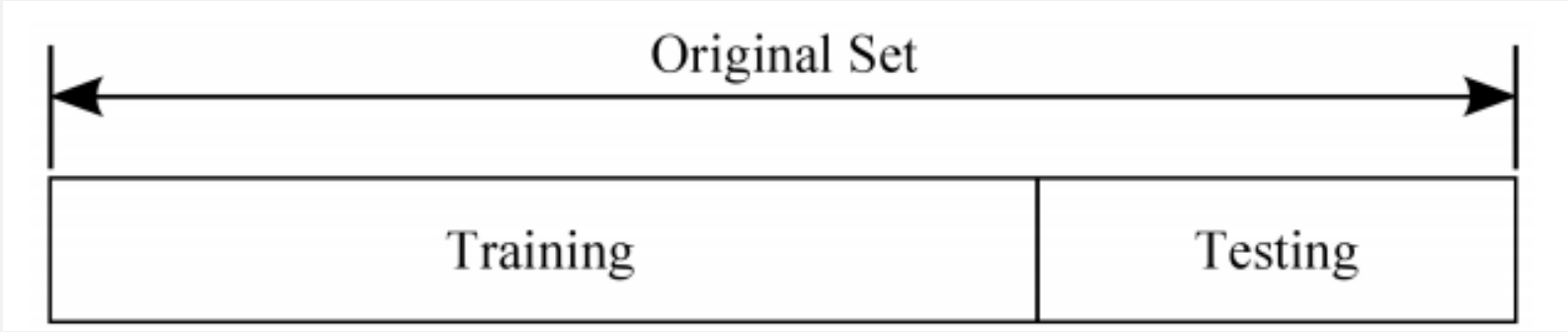
- 학습을 할 때 입력 값(독립변수)만 제공하고, 입력 값 사이의 관계나 패턴을 학습하는 모델을 생성하여 입력 값에 정답을 부여하는 방법
- 대표적으로 주어진 데이터 입력 값의 패턴으로 여러 개의 클러스터를 생성 후, 입력 값이 속하는 클러스터(정답) 번호를 부여하는 클러스터링 방법이 있음
- 예시
  - 고객 성향 세그멘테이션
  - 공정 이상치 탐지
  - 영화 컨텐츠 카테고리 분류



# 학습 데이터 집합, 테스트 데이터 집합

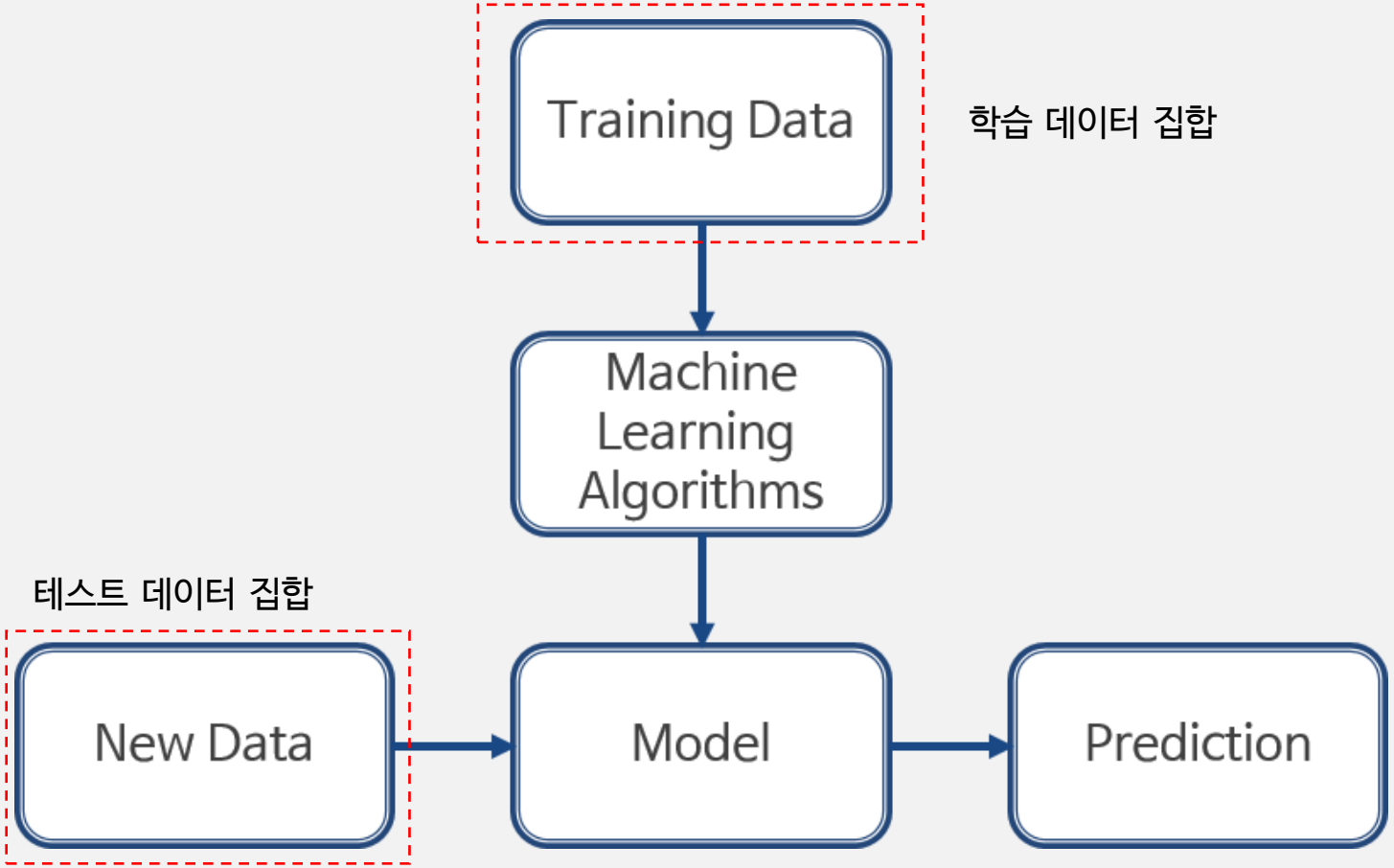
- 기계 학습 모델을 생성하기 위한 데이터 집합
  - 학습과 테스트 집합은 7:3, 8:2 등의 비율로 분리 가능
  - 학습과 테스트 집합은 모두 동일한 집합에서 추출되어야 함
  - 학습과 테스트 집합을 분리 시 경향(bias)이 없어야 함
- 학습 데이터 집합에서 만들어진 머신러닝 기계 학습 모델을 테스트 집합에 적합시켜 모델의 성능을 확인
  - 즉, 테스트 데이터 집합은 모델이 본 적이 없는 데이터 집합으로, 모델의 성능을 측정할 수 있는 기준 데이터 임

# 학습 데이터 집합, 테스트 데이터 집합




| 독립변수 | | 종속변수 | |
|---|---|---|---|
| 나이 | 성별 | 가입 | 구분 |
| 23 | 여자 | 1 | 학습 데이터 집합 |
| 55 | 남자 | 1 | 학습 데이터 집합 |
| 45 | 남자 | 1 | 학습 데이터 집합 |
| … | … | … | … |
| 55 | 여자 | 1 | 테스트 데이터 집합 |
| 17 | 남자 | 0 | 테스트 데이터 집합 |

# 학습 데이터 집합, 테스트 데이터 집합



# 모델 성능 - 지도학습의 분류 모델 성능 평가, Confusion Matrix

**예측**

**실제**

| | P | N |
|---|---|---|
| P | TP | FN |
| N | FP | TN |

**P : Positive(예측하고자 하는 범주)**
**N : Negative(기타 범주)**
**T : True (참)**
**F : False (거짓)**

**정밀도 (Precision)** = $\frac{TP}{TP + FP}$ **재현율 (Recall)** = $\frac{TP}{TP + FN}$

**정확도 (Accuracy)** = $\frac{TP + TN}{TP + TN + FP + FN}$

# 모델 성능 - 지도학습의 분류 모델 성능 평가

| | | 예측 | |
|---|---|---|---|
| | | P | N |
| 실제 | P | TP | FN |
| | N | FP | TN |

**(예시)**
**P : Positive (보험 사기 청구)**
**N : Negative (보험 사기 청구 아님)**

**정밀도(Precision) =** $\frac{TP}{TP+FP} = \frac{\text{실제 보험 사기 이며, 보험 사기로 예측한 청구 건수}}{\text{보험 사기로 예측한 청구 건수}}$

**재현율 (Recall) =** $\frac{TP}{TP+FN} = \frac{\text{실제 보험 사기 이며, 보험 사기로 예측한 청구 건수}}{\text{실제 보험 사기인 청구 건수}}$

**정확도 (Accuracy) =** $\frac{TP+TN}{TP+TN+FP+FN} = \frac{\text{정확하게 예측한 청구 건수}}{\text{전체 청구 건수}}$

# 모델 성능 - 지도학습의 분류 모델 성능 평가

| 실제 \ 예측 | P | N |
|---|---|---|
| P | TP 1500 | FN 1200 |
| N | FP 900 | TN 1100 |

- 실제 보험사기 이면서 보험 사기로 잘 예측한 건수 : 1,500 건
- 실제 보험사기지만 보험사기가 아닌 것으로 잘못 예측한 건수 : 1,200 건
- 실제 보험사기가 아니지만 보험사기로 잘못 예측한 건 : 900 건
- 실제 보험사기가 아니고 보험사기가 아닌 것으로 잘 예측한 건 : 1,100 건

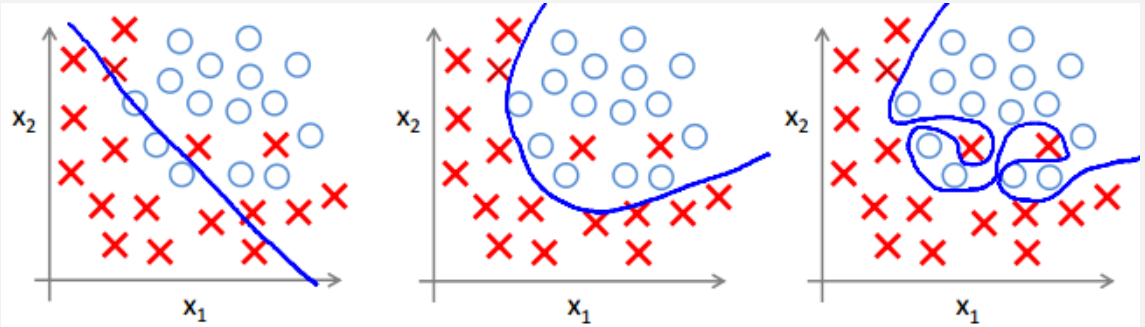
**정밀도(Precision)** = $\frac{TP}{TP + FP} = \frac{1{,}500}{1{,}500 + 900} = 62.5\%$

**재현율 (Recall)** = $\frac{TP}{TP + FN} = \frac{1{,}500}{1{,}500 + 1{,}200} = 55.5\%$

**정확도 (Accuracy)** = $\frac{TP + TN}{TP + TN + FP + FN} = \frac{1{,}500 + 1{,}100}{1{,}500 + 1{,}100 + 1{,}200 + 900} = \mathbf{55.3\%}$

# 과적합 (Overfitting)

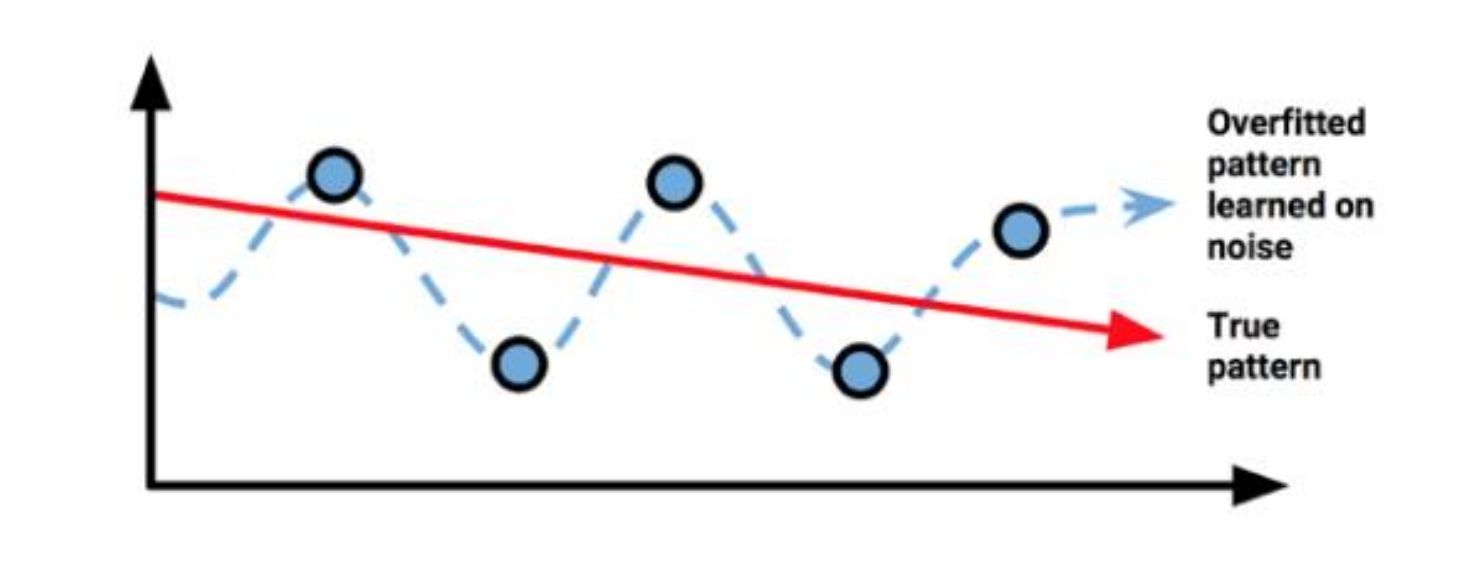
- 과적합
  - 학습 데이터 집합을 과하게 학습하여, 모델을 생성한 학습 데이터 집합에서의 성능대비 테스트 데이터 집합에서 그 성능이 낮은 경우
  - 학습 데이터 집합 기반 분류 모델 생성




| | | | |
|---|---|---|---|
| 학습 데이터 집합 | 80.1% | 94.3% | 97.3% |
| 테스트 데이터 집합 | 78.5% | 94.1% | 81.1% |

# 과적합 (Overfitting)

- 머신 러닝 모델은 학습데이터 집합을 기반으로 모델이 생성되고,
  모델이 본 적이 없는 테스트 데이터 집합에 적용하여
  그 성능을 평가함



# 법적, 윤리적 문제

- 사생활 (Privacy) 문제
  - 점점 더 중요해 짐
  - 데이터가 사용되고 분석되는 방법에 영향을 끼침
  - 소유권 이슈
- 특정 산업 부문의 정부 규제
  - 의약품 개발 과정 중 생기는 데이터 무결성 및 추적성에 대한 FDA (식품 의약국) 규정
- 종종 데이터웨어 하우스에 포함 된 데이터는 의사 결정 프로세스에서 합법적으로 사용할 수 없음
  - 인종, 성별, 연령 등
- 데이터 오염은 중요한 이슈

# Contents

R 

- S Language : 1976년 Bell Lab의 John Chambers, Rick Becker, Allan Wilks 에 의해 개발된 데이터 분석 언어
- R : 1993년에 S에서 영향을 받아 University of Auckland의 Ross Ihaka와 Robert Gentleman이 개발한 Open Source Software (R 이름의 유래: 두 R 개발자 이름의 첫 글자, “S” 이름을 계승)
- 1997년 이후: 15명으로 이루어진 international R-core team 을 기본으로, 1000 명 이상의 개발자와 분석가가 알고리즘을 패키지로 개발하여 기여

# R 이란? (1/2)

- R은 통계적 계산과 그래픽을 위한 언어이며 개발 환경
  - 통계 분석과 그래픽 작성을 위한 프로그래밍 언어
    - 통계학자에 의한, 통계학자를 위한 언어
  - 분석소프트웨어
    - 데이터 입출력, 데이터 처리, 데이터 분석, 그래프 작성 등을 위한 수많은 알고리즘 및 방법론 제공
    - 자체 개발 환경 제공

# R이란? (2/2)

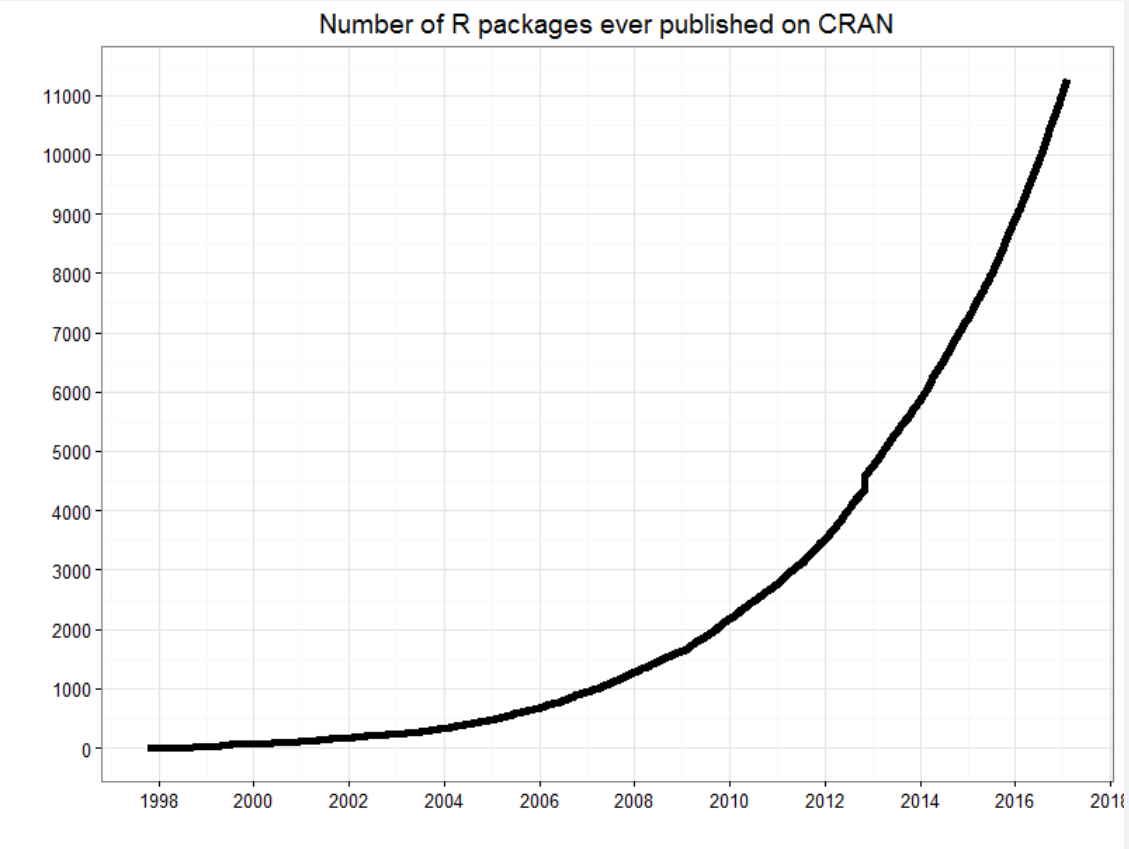
- R은 무료 소프트웨어
  - GNU(GNU's Not Unix!) GPL License
  - 사용자의입장에서 Free 라는 것이 단지 무료임을 뜻하는 것은 아님
  - 언제 어디서든 다운로드 및 설치가 가능
  - Windows, Linux, Unix, Mac 등 다양한 운영체제에서 동작
  - 누구나 패키지(Package)를 만들어 다른 사람과 공유 가능
  - Java, Python, .Net, Visual Studio, C, C++ 등 다양한 개발 언어 및 플랫폼과 연동

# R의 특징

- In-Memory Computing
  - 빠른 처리 속도
  - H/W 메모리 크기에 영향을 받음
- Object-oriented programming
  - 데이터, 함수가 object로 관리되어 짐
  - 클래스(class) & 메소드(method)를 가짐
- Package
  - 개인이 만들어서 등록할 수 있는 R 함수
  - 최신의 알고리즘 및 방법론을 적용
  - 다양한 함수 및 데이터 내장

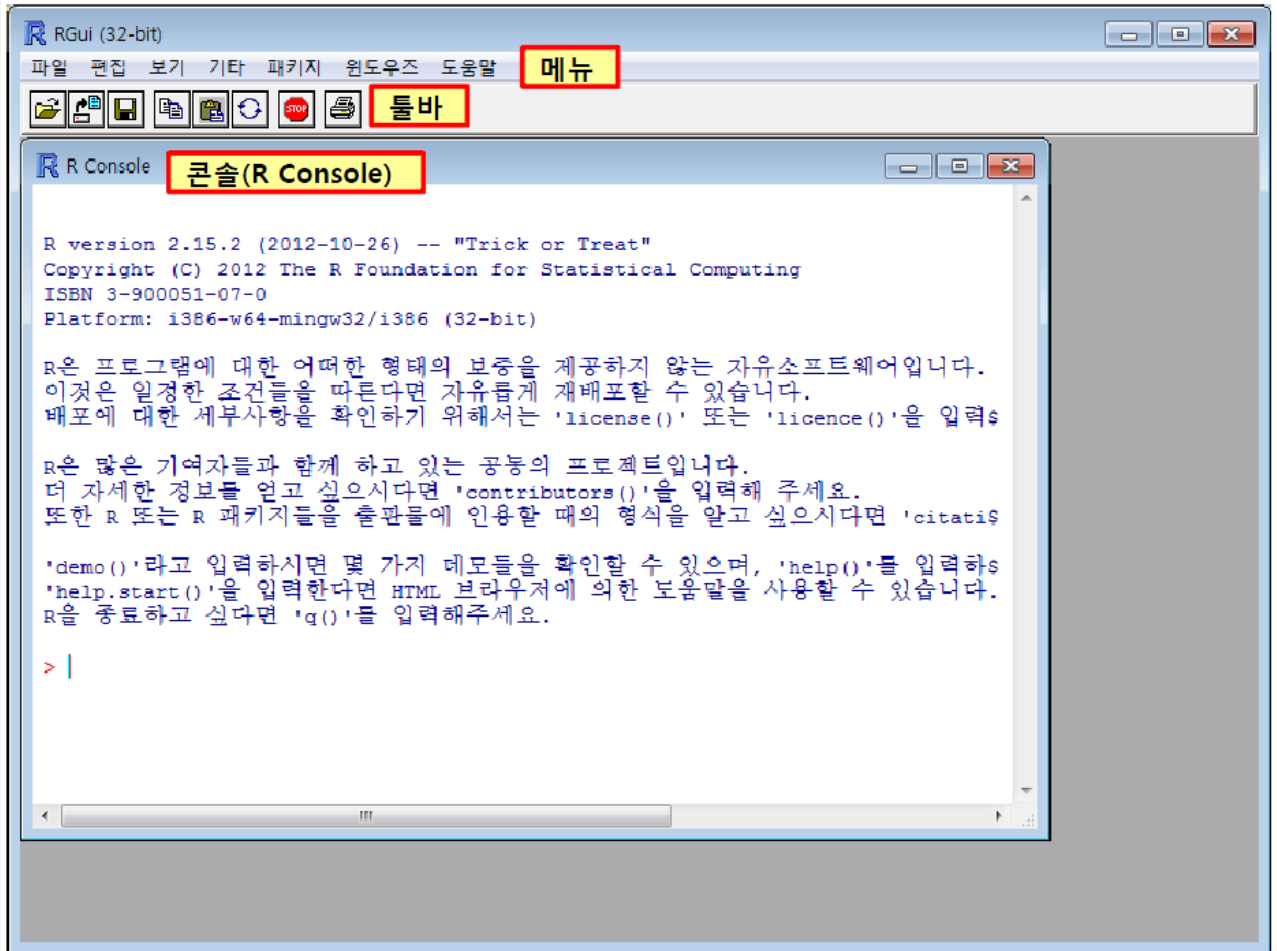
# R Packages

- R Package (in CRAN)
  - CRAN (The Comprehensive R Archive Network)
  - CRAN Site에 12,744개 등록 됨 (2018년 7월 기준)
  - 새로운 통계 분석 알고리즘이나 새로운 IT 기술의 응용에 관한 것을 포함
  - Software Vendor에 의하여 Version Up 되지 않는다는 것이 다른 통계 분석 소프트웨어와의 차이점 임



# R GUI (1/2)

- R Gui 실행 기본  화면은 메뉴, 툴바 (단축아이콘), 콘솔창으로 구성



# R GUI (2/2)

- 입력된 명령(command)에 대한 결과가 interactive하게 화면에 출력된다.

```
> getwd()
[1] "C:/Users/65795/Documents"
> search()
[1] ".GlobalEnv"        "package:stats"     "package:graphics"  "package:grDevices" "package:utils"     "package:datasets"  "package:methods"   "Autoloads"         "package:base"
> searchpaths()
[1] ".GlobalEnv"                                "C:/Program Files/R/R-2.15.2/library/stats"     "C:/Program Files/R/R-2.15.2/library/graphics"  "C:/Program Files/R/R-2.15.2/library/grDevices"
[5] "C:/Program Files/R/R-2.15.2/library/utils"     "C:/Program Files/R/R-2.15.2/library/datasets"  "C:/Program Files/R/R-2.15.2/library/methods"   "Autoloads"
[9] "C:/PROGRA~1/R/R-215~1.2/library/base"
> ls
function (name, pos = -1, envir = as.environment(pos), all.names = FALSE,
    pattern)
{
    if (!missing(name)) {
        nameValue <- try(name, silent = TRUE)
        if (identical(class(nameValue), "try-error")) {
            name <- substitute(name)
            if (!is.character(name))
                name <- deparse(name)
            warning(sQuote(name), " converted to character string")
            pos <- name
        }
        else pos <- nameValue
    }
    all.names <- .Internal(ls(envir, all.names))
    if (!missing(pattern)) {
        if ((ll <- length(grep("[", pattern, fixed = TRUE))) &&
            ll != length(grep("]", pattern, fixed = TRUE))) {
            if (pattern == "[") {
                pattern <- "\\["
                warning("replaced regular expression pattern '[' by  '\\\\['")
            }
            else if (length(grep("[^\\\\]\\[<-", pattern))) {
                pattern <- sub("\\[<-", "\\\\[<-", pattern)
                warning("replaced '[<-' by '\\\\[<-' in regular expression pattern")
            }
        }
        grep(pattern, all.names, value = TRUE)
    }
    else all.names
}
<bytecode: 0x05e68150>
<environment: namespace:base>
> ls()
character(0)
> ls(pos=6)
  [1] "ability.cov"       "airmiles"          "AirPassengers"     "airquality"        "anscombe"          "attenu"            "attitude"          "austres"
  [9] "beaver1"           "beaver2"           "BJsales"           "BJsales.lead"      "BOD"               "cars"              "ChickWeight"       "chickwts"
 [17] "co2"               "CO2"               "crimtab"           "discoveries"       "DNase"             "esoph"             "euro"              "euro.cross"
 [25] "eurodist"          "EuStockMarkets"    "faithful"          "fdeaths"           "Formaldehyde"      "freeny"            "freeny.x"          "freeny.y"
 [33] "HairEyeColor"      "Harman23.cor"      "Harman74.cor"      "Indometh"          "infert"            "InsectSprays"      "iris"              "iris3"
 [41] "islands"           "JohnsonJohnson"    "LakeHuron"         "ldeaths"           "lh"                "LifeCycleSavings"  "Loblolly"          "longley"
 [49] "lynx"              "mdeaths"           "morley"            "mtcars"            "nhtemp"            "Nile"              "nottem"            "occupationalStatus"
 [57] "Orange"            "OrchardSprays"     "PlantGrowth"       "precip"            "presidents"        "pressure"          "Puromycin"         "quakes"
 [65] "randu"             "rivers"            "rock"              "Seatbelts"         "sleep"             "stack.loss"        "stack.x"           "stackloss"
 [73] "state.abb"         "state.area"        "state.center"      "state.division"    "state.name"        "state.region"      "state.x77"         "sunspot.month"
```

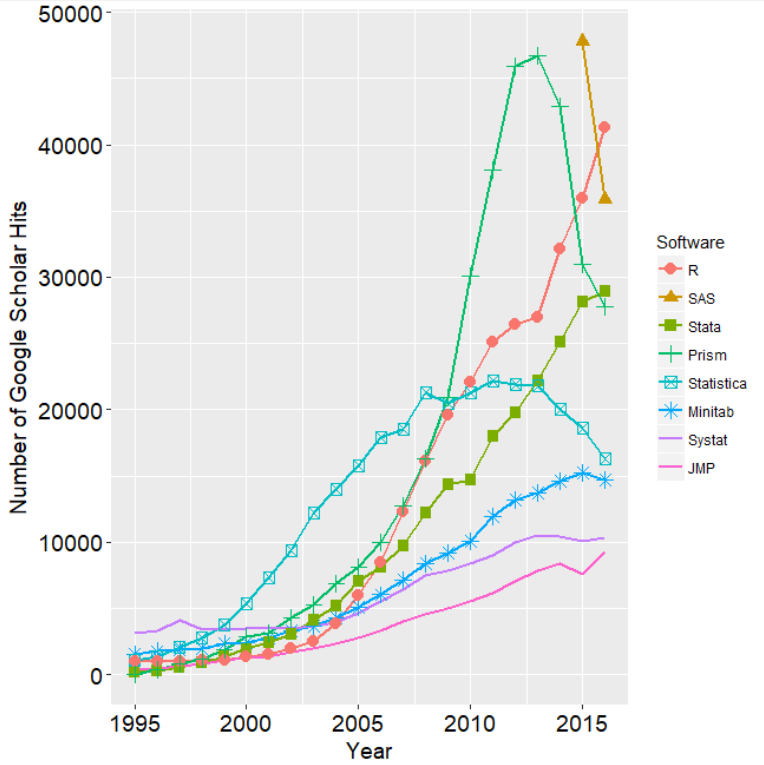
❑ **R Console에 입력된 R command**

- **getwd()　: 현재 working directory 확인**
- **search() : R object와 package 리스트**
- **searchpaths() : R package가 존재하는 path**
- **ls : 오프젝트 리스트를 문자열로 보여주는 함수**
- **ls() : ls 함수 실행**
- **ls(pos=6) : search() 결과의 6번째 패키지내의 object 리스트**

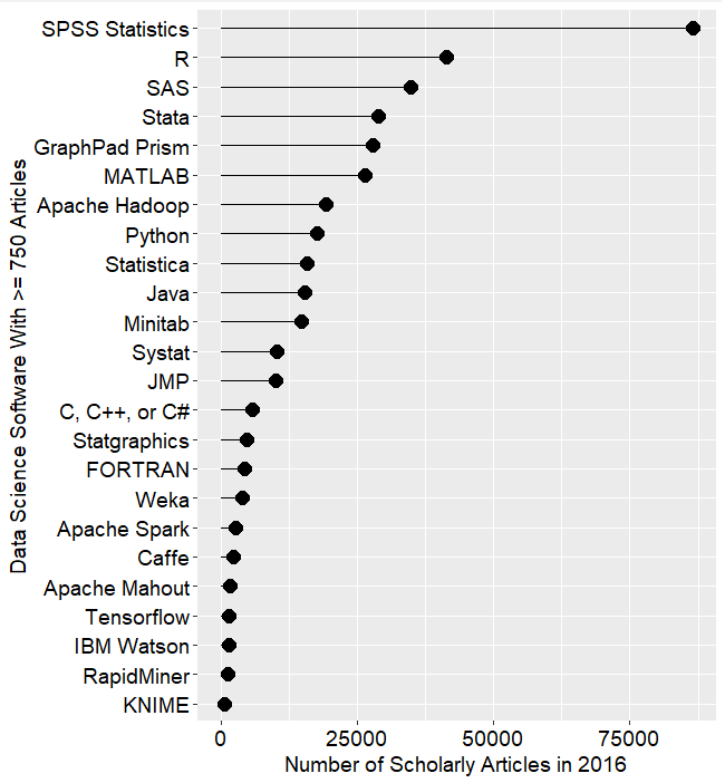
# Usage of R (1/2)

## 분석 패키지별 학술연구 활용 빈도

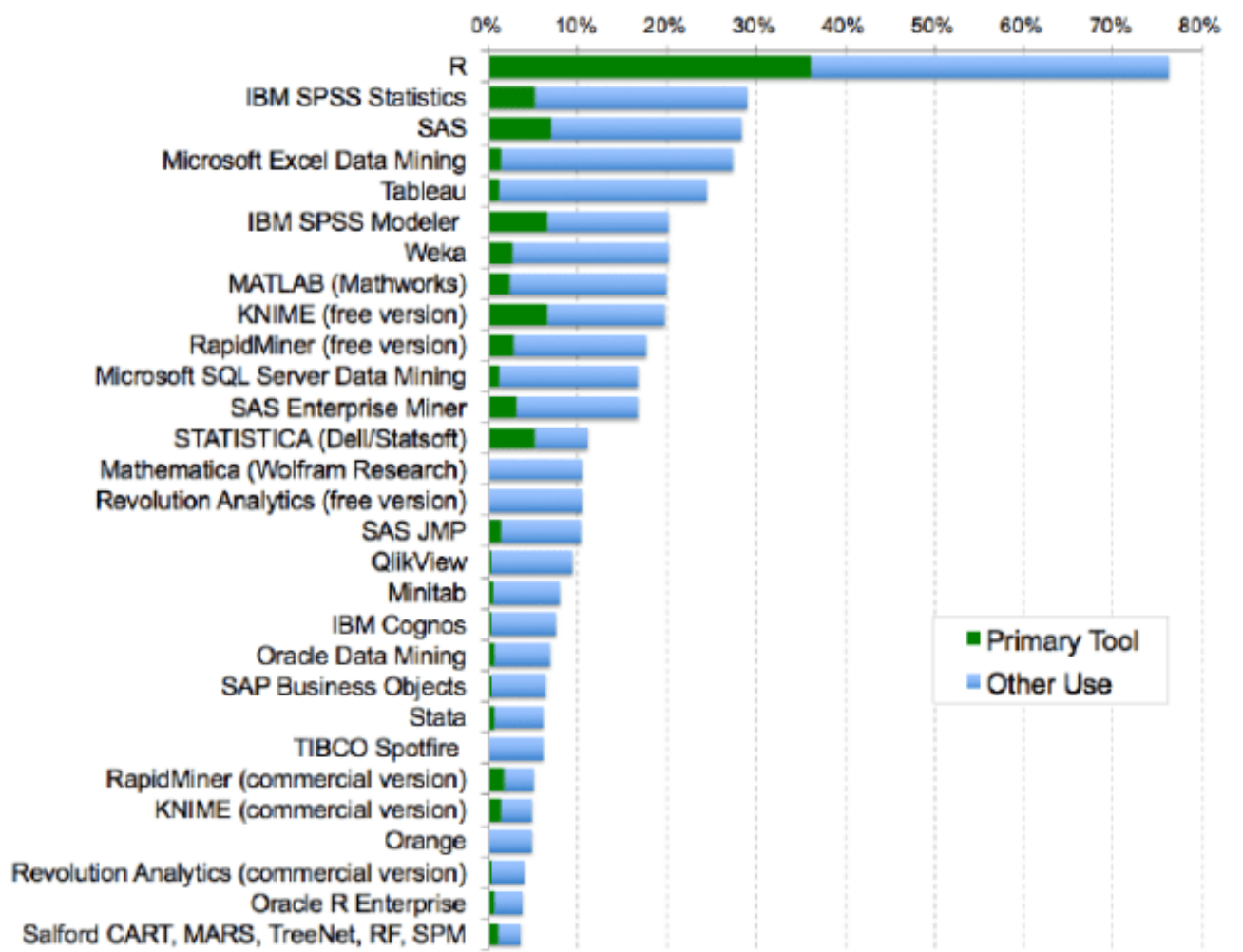
## (www.r4stats.com)

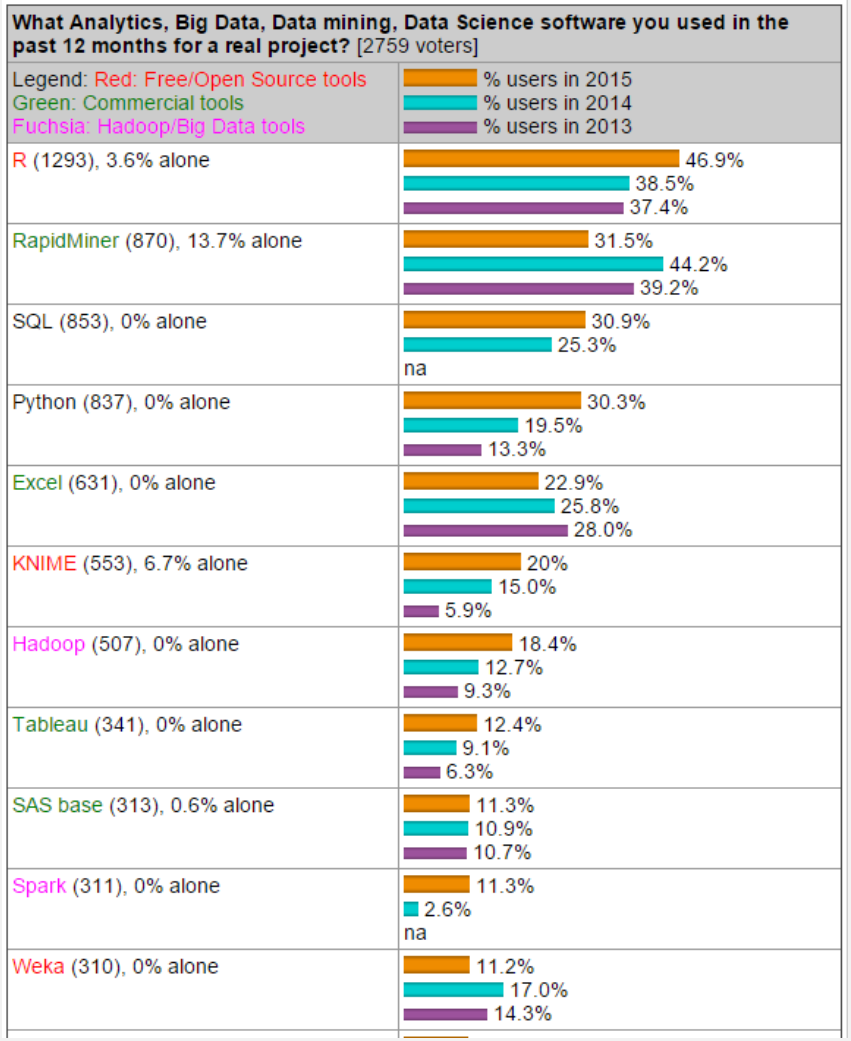



## 최근 논문에 사용된 소프트웨어



# Usage of R (2/2)



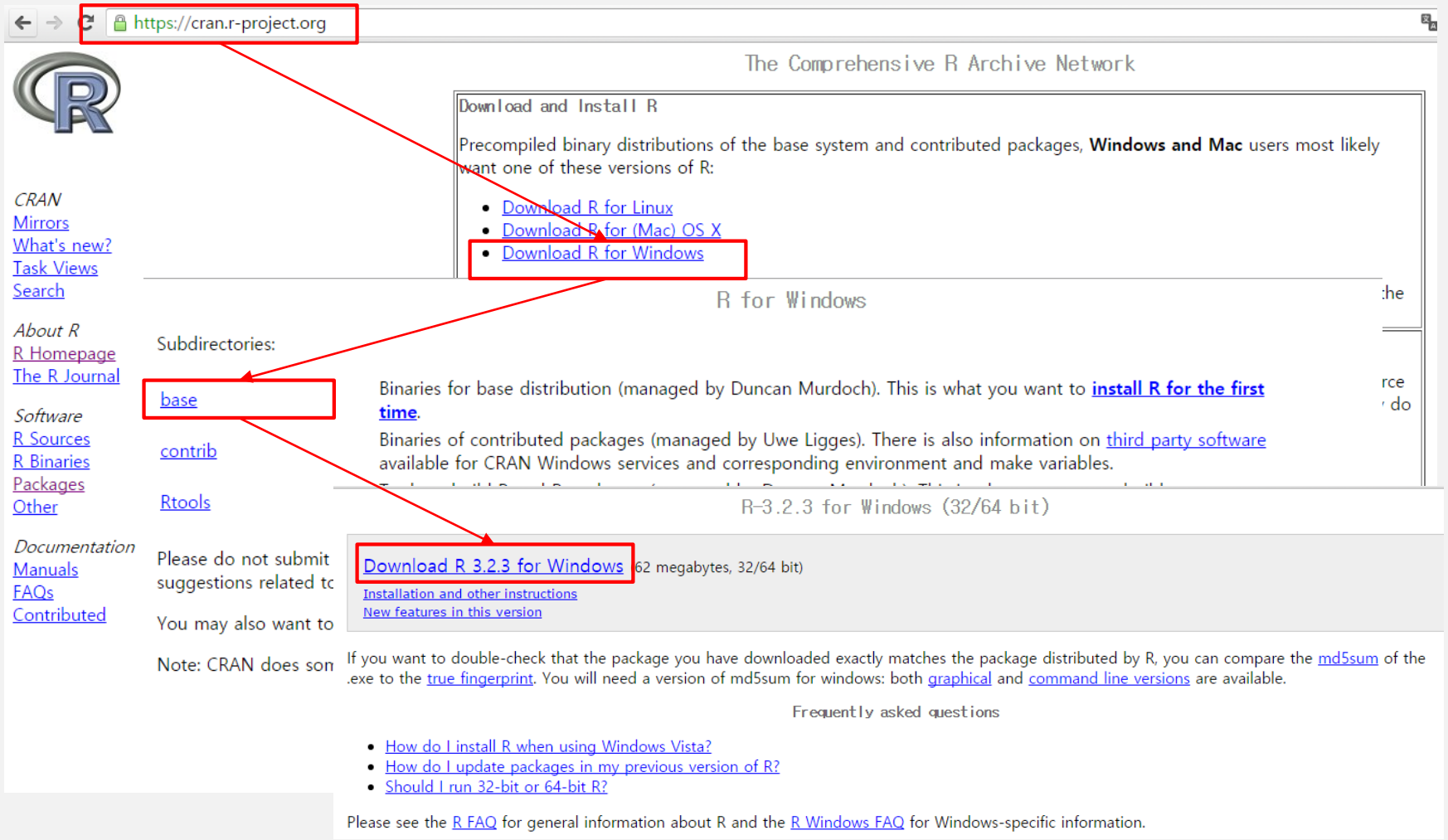




출처: kdnuggets.com

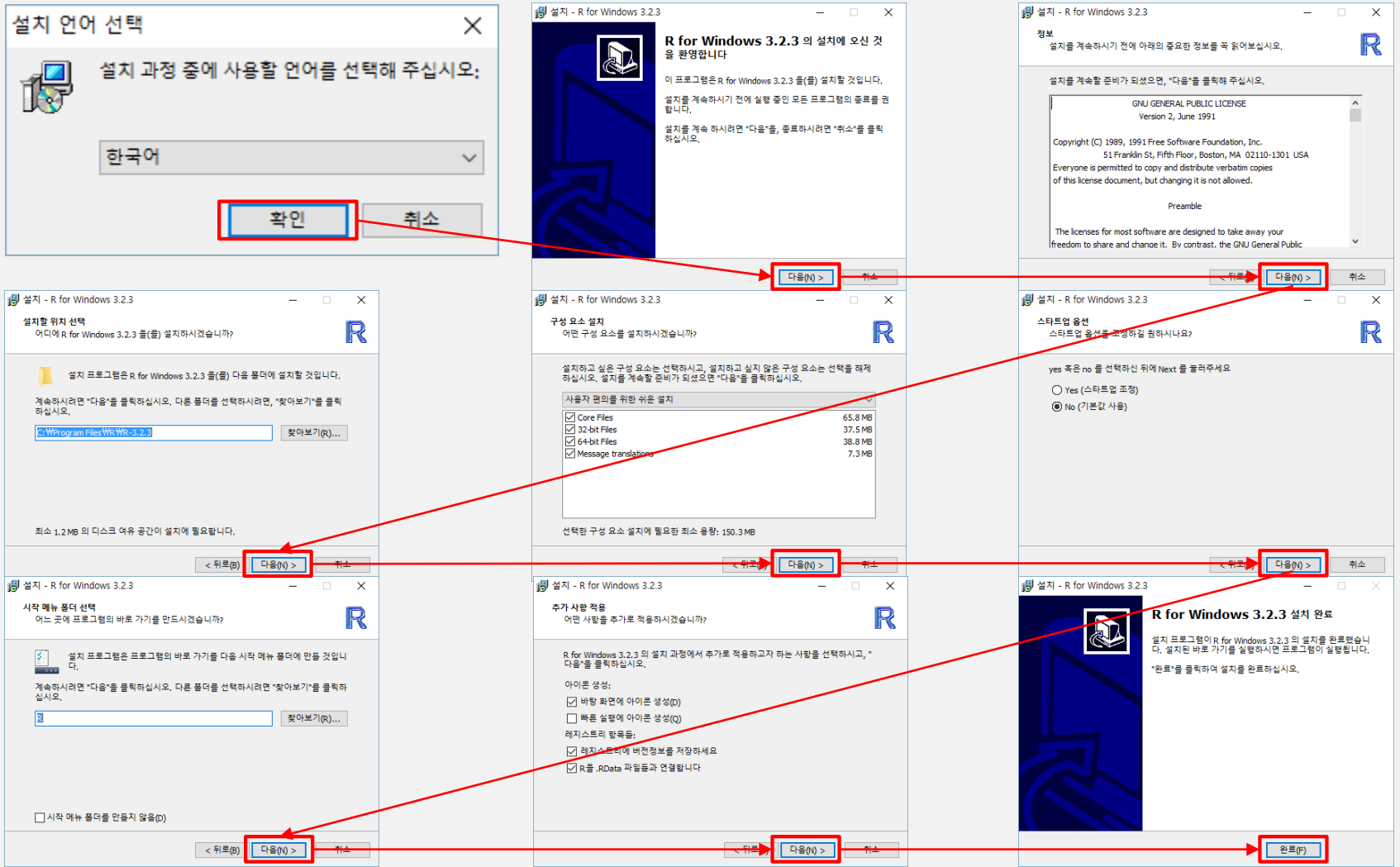
# 이슈

- 컴퓨터 프로그래밍 언어로 전문가에게는 강력한 도구이지만, 일반 사용자가 사용하기에는 진입장벽이 존재한다.
  - R Community에 다양한 GUI 관련 Project 존재
- 분석 가능한 데이터 크기는 in-memory에서 실행 가능한 크기로 제한된다.
  - 32bit Machine - $\frac{2^{32}}{1024^3} = 4GB$,
  - 64bit machine - Windows : RAM 크기-Linux : 이론적으로는 무한하나 Disk Swap으로 성능 저하
  - 메모리 제약을 극복하는 Package가 있으나, 분석 알고리즘은 개발 필요
- 대용량 데이터 집합은 R의 수행 시간을 저하시킨다.
  - R은 CPU에서 Single Core만 사용, 데이터가 적으면 괜찮으나, 많으면 수행 시간이 느려짐
  - 병렬 처리 할 수 있는 Package가 있으나, 분석 알고리즘은 개발 필요
- 분석 모델 개발과 별개로 분석 모델을 적용/관리하는 기능에 대한 개발은 필요하다.
  - 스케줄실행, 외부실행, 어플리케이션 연계, 사용자 관리 등

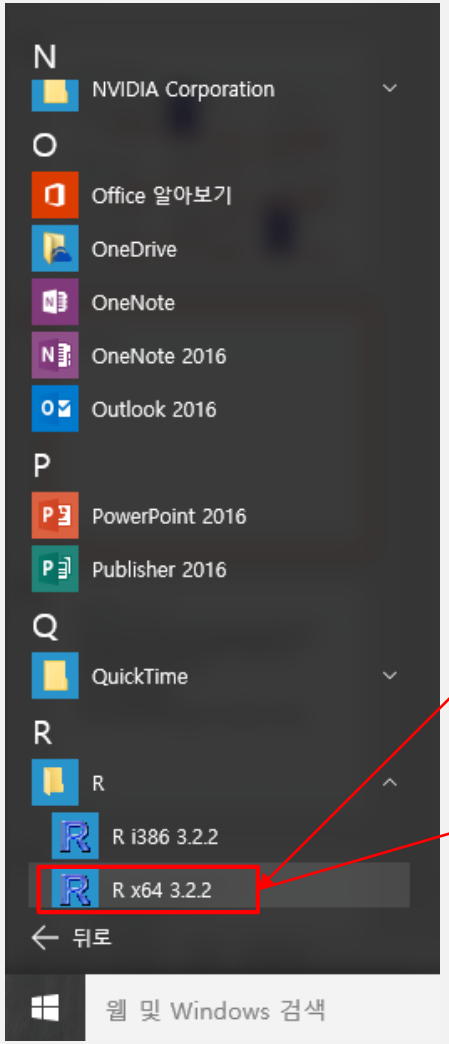
# R 설치 (1/2) http://cran.r-project.org

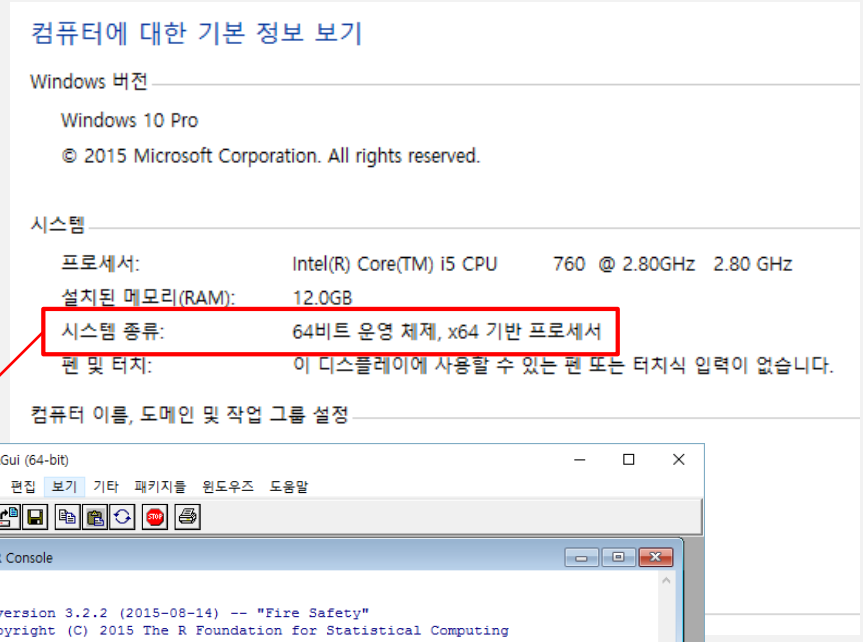


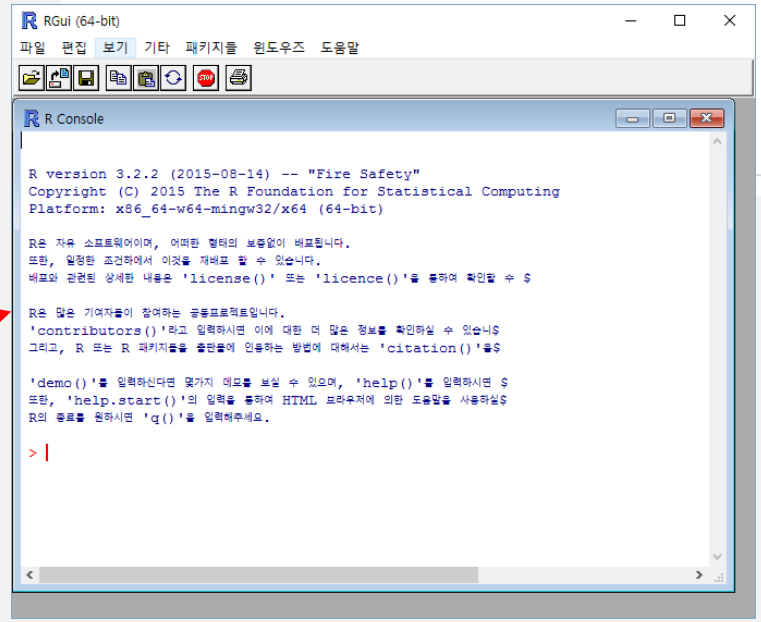
# R 설치 (2/2)


설치 언어 선택
설치 과정 중에 사용할 언어를 선택해 주십시오:
한국어
확인
취소
설치 - R for Windows 3.2.3
R for Windows 3.2.3 의 설치에 오신 것을 환영합니다
이 프로그램은 R for Windows 3.2.3 을(를) 설치할 것입니다.
설치를 계속하시기 전에 실행 중인 모든 프로그램의 종료를 권합니다.
설치를 계속 하시려면 "다음"을, 종료하시려면 "취소"를 클릭하십시오.
다음(N) >
취소
정보
설치를 계속하시기 전에 아래의 중요한 정보를 꼭 읽어보십시오.
설치를 계속할 준비가 되셨으면, "다음"을 클릭해 주십시오.
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
< 뒤로(B)
다음(N) >
취소
설치할 위치 선택
어디에 R for Windows 3.2.3 을(를) 설치하시겠습니까?
설치 프로그램은 R for Windows 3.2.3 을(를) 다음 폴더에 설치할 것입니다.
계속하시려면 "다음"을 클릭하십시오. 다른 폴더를 선택하시려면, "찾아보기"를 클릭하십시오.
C:\Program Files\R\R-3.2.3
찾아보기(R)...
최소 1.2 MB 의 디스크 여유 공간이 설치에 필요합니다.
구성 요소 설치
어떤 구성 요소를 설치하시겠습니까?
설치하고 싶은 구성 요소는 선택하시고, 설치하고 싶지 않은 구성 요소는 선택을 해제하십시오. 설치를 계속할 준비가 되셨으면 "다음"을 클릭하십시오.
사용자 편의를 위한 쉬운 설치
Core Files 65.8 MB
32-bit Files 37.5 MB
64-bit Files 38.8 MB
Message translations 7.3 MB
선택한 구성 요소 설치에 필요한 최소 용량: 150.3 MB
스타트업 옵션
스타트업 옵션을 지정하길 원하시나요?
yes 혹은 no 를 선택하신 뒤에 Next 를 눌러주세요
Yes (스타트업 조정)
No (기본값 사용)
시작 메뉴 폴더 선택
어느 곳에 프로그램의 바로 가기를 만드시겠습니까?
설치 프로그램은 프로그램의 바로 가기를 다음 시작 메뉴 폴더에 만들 것입니다.
계속하시려면 "다음"을 클릭하십시오. 다른 폴더를 선택하시려면 "찾아보기"를 클릭하십시오.
찾아보기(R)...
시작 메뉴 폴더를 만들지 않음(D)
추가 사항 적용
어떤 사항을 추가로 적용하시겠습니까?
R for Windows 3.2.3 의 설치 과정에서 추가로 적용하고자 하는 사항을 선택하시고, "다음"을 클릭하십시오.
아이콘 생성:
바탕 화면에 아이콘 생성(D)
빠른 실행에 아이콘 생성(Q)
레지스트리 항목들:
레지스트리에 버전정보를 저장하세요
R을 .RData 파일들과 연결합니다
R for Windows 3.2.3 설치 완료
설치 프로그램이 R for Windows 3.2.3 의 설치를 완료했습니다. 설치된 바로 가기를 실행하시면 프로그램이 실행됩니다.
"완료"를 클릭하여 설치를 완료하십시오.
완료(F)

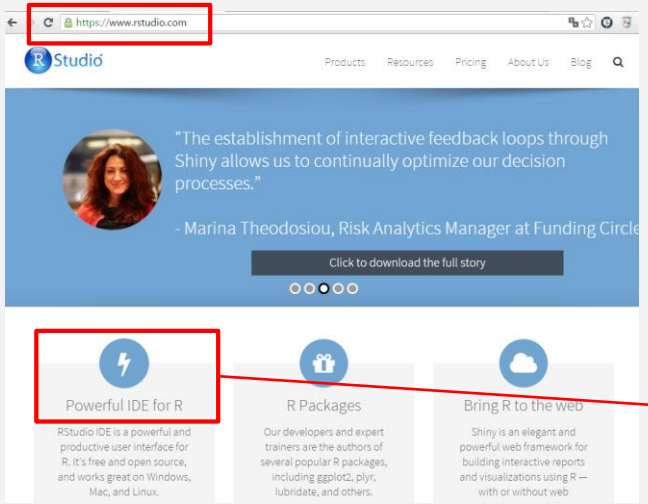
# R 실행

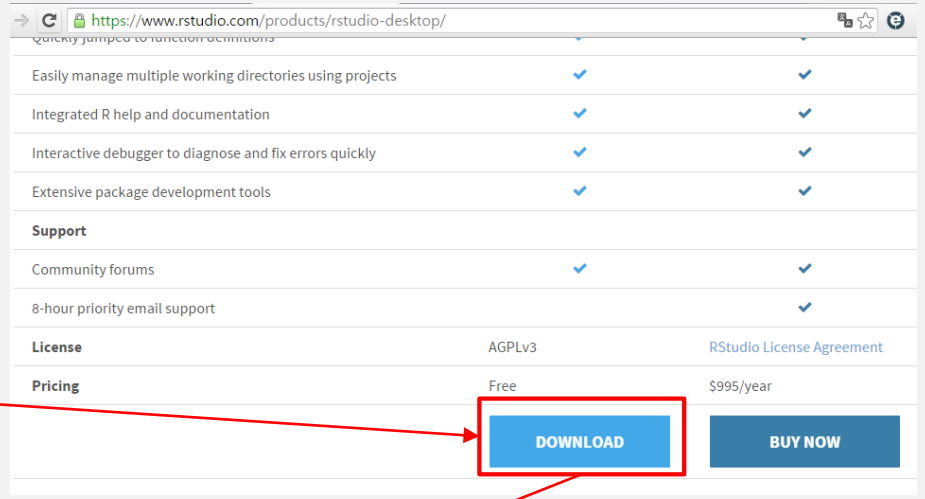

N
NVIDIA Corporation
O
Office 알아보기
OneDrive
OneNote
OneNote 2016
Outlook 2016
P
PowerPoint 2016
Publisher 2016
Q
QuickTime
R
R
R i386 3.2.2
R x64 3.2.2
뒤로
웹 및 Windows 검색

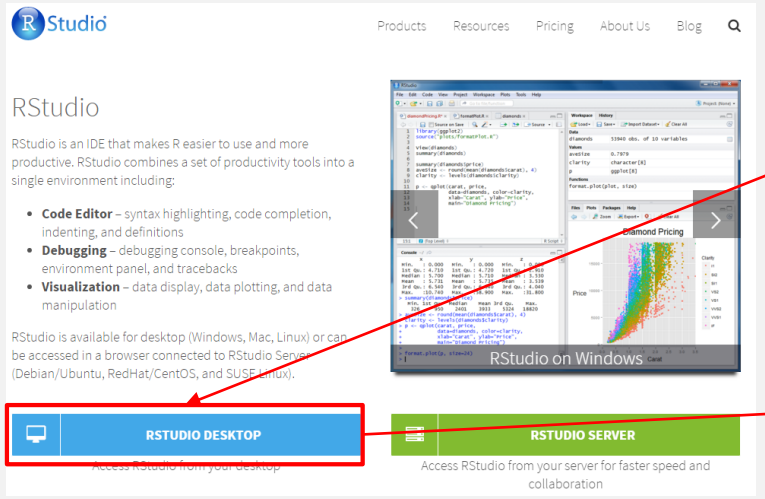

컴퓨터에 대한 기본 정보 보기
Windows 버전
Windows 10 Pro
© 2015 Microsoft Corporation. All rights reserved.
시스템
프로세서: Intel(R) Core(TM) i5 CPU 760 @ 2.80GHz 2.80 GHz
설치된 메모리(RAM): 12.0GB
시스템 종류: 64비트 운영 체제, x64 기반 프로세서
펜 및 터치: 이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다.
컴퓨터 이름, 도메인 및 작업 그룹 설정

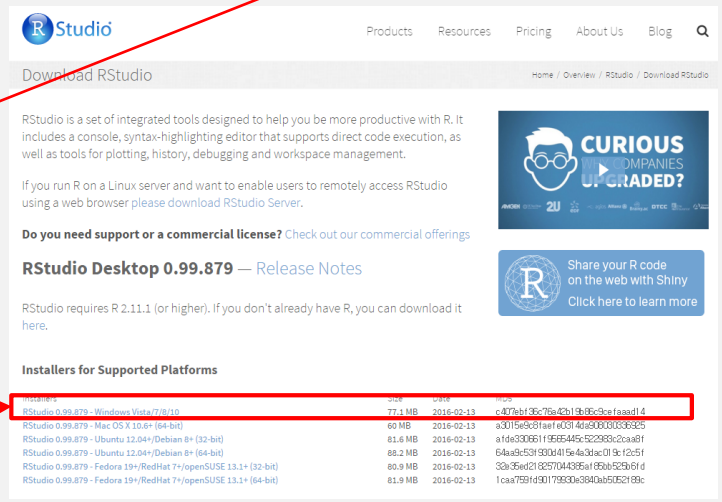

RGui (64-bit)
파일 편집 보기 기타 패키지들 윈도우즈 도움말
R Console
R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R은 자유 소프트웨어이며, 어떠한 형태의 보증없이 배포됩니다.
또한, 일정한 조건하에서 이것을 재배포 할 수 있습니다.
배포와 관련된 상세한 내용은 'license()' 또는 'licence()'을 통하여 확인할 수 $
R은 많은 기여자들이 참여하는 공동프로젝트입니다.
'contributors()'라고 입력하시면 이에 대한 더 많은 정보를 확인하실 수 있습니$
그리고, R 또는 R 패키지들을 출판물에 인용하는 방법에 대해서는 'citation()'을$
'demo()'를 입력하신다면 몇가지 데모를 보실 수 있으며, 'help()'를 입력하시면 $
또한, 'help.start()'의 입력을 통하여 HTML 브라우저에 의한 도움말을 사용하실$
R의 종료를 원하시면 'q()'을 입력해주세요.
>

# RStudio 설치(1/2) https://www.rstudio.com

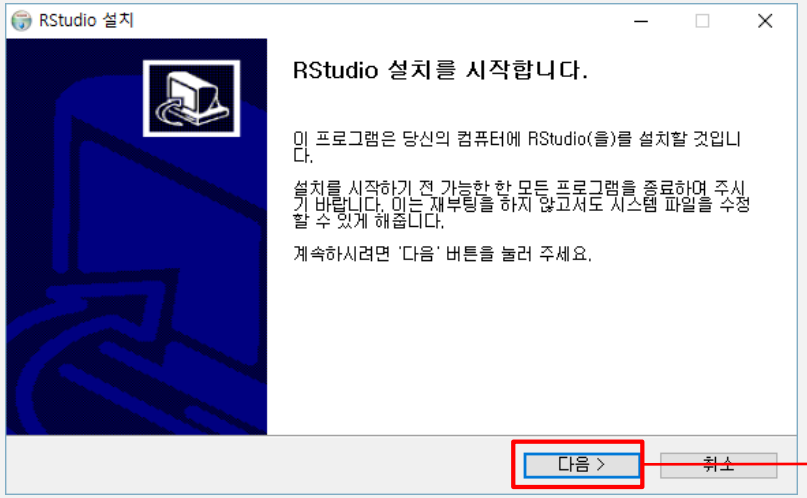


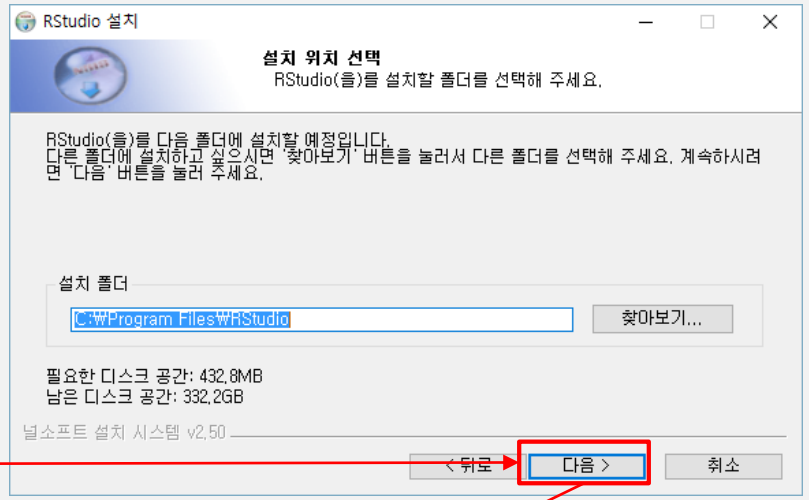



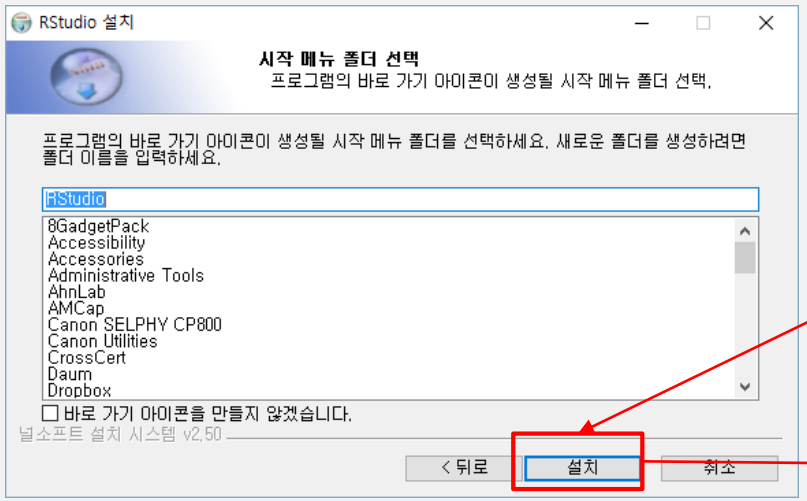



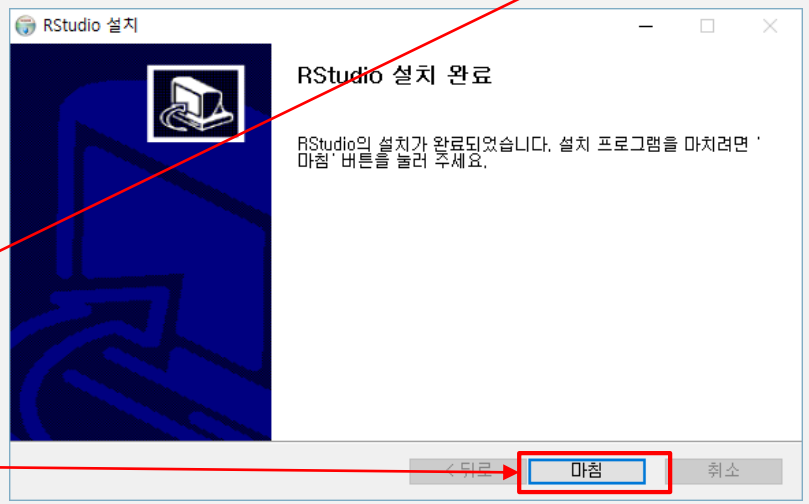



# RStudio 설치(2/2)

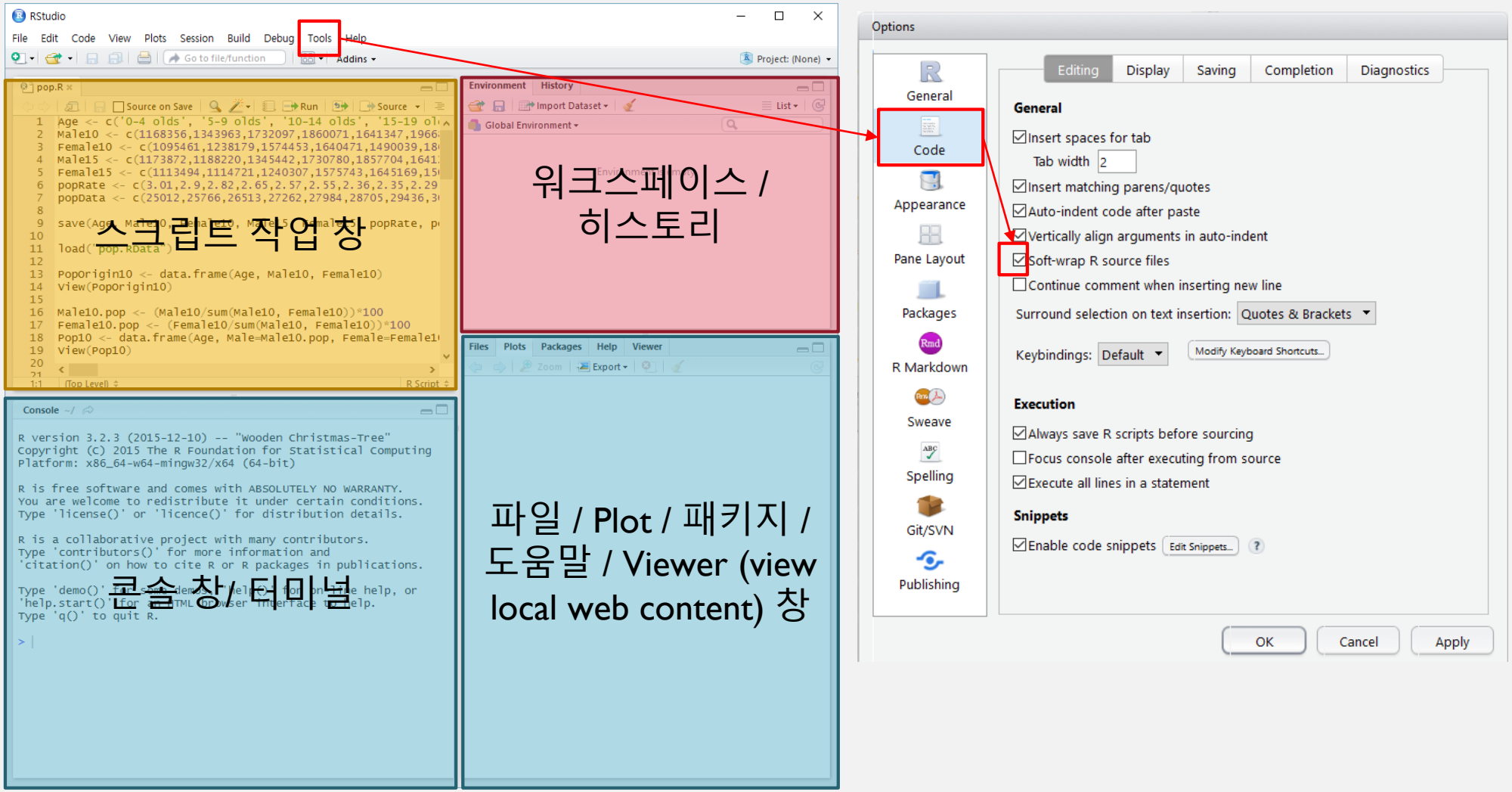


# RStudio 화면


RStudio
File Edit Code View Plots Session Build Debug Tools Help
스크립트 작업 창
워크스페이스 / 히스토리
콘솔 창/터미널
파일 / Plot / 패키지 / 도움말 / Viewer (view local web content) 창
Options
General
Code
Appearance
Pane Layout
Packages
R Markdown
Sweave
Spelling
Git/SVN
Publishing
Editing Display Saving Completion Diagnostics
Insert spaces for tab
Tab width 2
Insert matching parens/quotes
Auto-indent code after paste
Vertically align arguments in auto-indent
Soft-wrap R source files
Continue comment when inserting new line
Surround selection on text insertion: Quotes & Brackets
Keybindings: Default
Modify Keyboard Shortcuts...
Execution
Always save R scripts before sourcing
Focus console after executing from source
Execute all lines in a statement
Snippets
Enable code snippets
Edit Snippets...
OK
Cancel
Apply

# Contents

# 객체 지향 (Object-oriented) 프로그래밍 (1/2)

- 클래스 (Class)
  - 특정한 속성 (attribute)을 가지고 있는 객체(object)의 추상적인 정의
  - Dog 클래스
    - 속성 : color, size, age
- 객체 (Object)
  - 특정 클래스의 instance (구현된 것)
    - Dog1 – color: ‘brown’, size: 5.5 inch, age: 3 years
- 함수 (Function)
  - 입력 집합을 받아서 출력을 반환하는 절차의 집합
  - 값이 반환되는 것과 값이 반환되지 않는 것 존재

# 객체 지향 (Object-oriented) 프로그래밍 (2/2)

- 함수
  - 함수의 종류

```
var1 <- sum(c(4, 3, 2))
var1

var2 <- cat(9)
var2
```

  - 함수의 정의

```
test.function <- function(a, b, c){
  result <- (a * b) + c
  return(result)
}
test.function(2, 3, 1)
```

# 변수

- 자유로운 변수 할당
  - Overriding: 동일 변수에 다른 타입의 객체 할당 가능

```
var1 <- 10
var1 <- 'a string'
```

  - 변수의 타입 선언 불필요
  - 변수에 값 할당 방법
    - <- or ->: 연결된 값을 variable에 할당
    - =: <- 와 동일 (함수의 속성에 대한 치환 기호로 사용)
    - assign(): 첫번째 속성이 변수, 두번째 속성이 값
  - R은 대소문자 구분

# R 데이터 유형 종류

<table>
<tr><th>단일 데이터</th><th>1 차원</th><th>2 차원</th><th>n 차원</th></tr>
<tr><td>atomic</td><td>vector</td><td>matrix</td><td>array</td></tr>
<tr><td rowspan="3">character<br>numeric<br>integer<br>complex<br>logical<br>factor</td><td>c (1, 2, 3)</td><td>matrix<br>(c(1, 2, 3, 4), nrow = 2)</td><td rowspan="3">array<br>(c(1, 2, 3, 4, 5, 6),<br>dim = c(1, 2, 3))</td></tr>
<tr><td>list</td><td>data.frame</td></tr>
<tr><td>list (a = “1”, b = 2)</td><td>data.frame<br>(id = c(1, 2, 3),<br>da = c(‘aa’, ‘bb’,+ ‘cc’))</td></tr>
</table>

# 데이터 타입 클래스 (1/6)

- 단일 데이터 타입 클래스
  - Character: 문자열, 따옴표로 구분
  - Numeric: 소수 숫자
  - Integer: 정수 숫자
  - Complex: 복소수
  - Logical: TRUE/FALSE
  - Factor: Enum, 카테고리화

# 데이터 타입 클래스 (2/6)

- Vectors
  - 하나의 클래스 만 가지는 원소(element)들의 집합
  - Vector의 타입 (type)은 원소의 타입과 동일
  - 기존 타입과 다른 원소가 추가되면, 에러가 아닌, vector의 타입을 변경

```
aaa <- numeric(length = 5)
aaa[1] <- 6
aaa[2] <- 2
class(aaa)
aaa[3] <- 'a string'
class(aaa)
aaa[1] - aaa[2]
```

# 데이터 타입 클래스 (3/6)

- Lists
  - 어떤 클래스의 어떤 객체도 포함하는 vector
  - List는 list를 포함할 수 있음

```
aaa <- list()
aaa[1] <- 4
aaa[2] <- 5
aaa[3] <- 'a string'
aaa
aaa[[2]] - aaa[[1]]
```

# 데이터 타입 클래스 (4/6)

- Matrix and array
  - Vector의 한 종류: 차원(dimension) 속성을 가진 vector

```
numeric.vector <- 1:20
attr(numeric.vector, 'dim') <- c(10, 2)
class(numeric.vector)
```

  - Matrix는 2 차원 (열 (row), 행 (column))을 가지는 array의 한 종류

```
numeric.vector <- 1:20
numeric.vector <- matrix(numeric.vector, 10, 2)
class(numeric.vector)
```

  - Vector와 동일하게 동일 타입의 원소들만 가질 수 있음

# 데이터 타입 클래스 (5/6)

- Data frames
  - List의 특별한 형태: List의 원소들의 길이가 동일
  - 2 차원 matrix와 유사: 다른 클래스들을 원소로 가질 수 있는 것이 다름

```
numeric.vector <- 1:5
character.vector <- letters[1:5]
class(numeric.vector)
class(character.vector)
df <- data.frame(numeric.vector, character.vector)
class(df)
```

# 데이터 타입 클래스 (6/6)

- Factors
  - 범주형 (Categorical) 변수를 위한 특별한 클래스
  - 문자 원소를 나타내는 숫자 코드
  - Integer의 집합과 연관된 level (label)로 이루어짐

```
animals <- c('dog', 'cat', 'dog', 'horse')
class(animals)
animals
animals <- as.factor(animals)
animals
cat(animals)
as.character(animals)
as.numeric(animals)
```

  - 자동 변환 주의 : stringsAsFactors=FALSE

# Vector의 이름

- names
  - Vector의 원소들에 이름 부여

```
Poker_vector <- c(40, 150, -30, 20, -240)
names(Poker_vector)
names(Poker_vector) <- c('Monday', 'Tuesday', 'Wednesday','Thursday', 'Friday')
names(Poker_vector)
```

# Vector의 연산 (1/3)

- +, -
  - Vector의 원소 끼리의 덧셈, 뺄셈

```
VecA <- c(1, 2, 3)
VecB <- c(4, 5, 6)

Total_vec <- VecA + VecB
Total_vec

Diff_vec <- VecA – VecB
Diff_vec
```

# Vector의 연산 (2/3)

- *, /, %/%, %%
  - Vector의 원소 끼리의 곱셈, 나눗셈, 몫, 나머지

```
VecA <- c(1, 2, 3)
VecB <- c(4, 5, 6)

Mul_vec <- VecB * VecA
Mul_vec
Div_vec <- VecB / VecA
Div_vec
Qu_vec <- VecB %/% VecA
Qu_vec
Re_vec <- VecB %% VecA
Re_vec
```

# Vector의 연산 (3/3)

- sum, mean
  - Vector 원소의 합 / 평균

```
Poker_vector <- c(40, 150, -30, 20, -240)

TotalPoker <- sum(Poker_vector)
TotalPoker
MeanPoker <- mean(Poker_vector)
MeanPoker
```

# 행렬(Matrix)의 연산 (1/4)

- *, /
  - 행렬과 상수의 곱셈, 나눗셈

```
x <- matrix(c(1:9), ncol = 3)
x
x * 2
x / 2
```

# 행렬(Matrix)의 연산 (2/4)

- +, -
  - 크기가 동일한 행렬과 행렬의 덧셈, 뺄셈

```
y <- matrix(c(9:1), nrow = 3)
x + y
x - y
```

# 행렬(Matrix)의 연산 (3/4)

- %*%
  - 행렬과 행렬의 곱셈
  - 앞 행렬의 열(column)의 수와 뒷 행렬의 행(row)의 수가 일치

x %*% y
x[1,1] * y[1,1] + x[1,2] * y[2,1] + x[1,3] * y[3,1]
1 * 9 + 4 * 8 + 7 * 7

# 행렬(Matrix)의 연산 (4/4)

- t( )
  - 행과 열을 바꾸는 전치 행렬
- ncol( ), nrow( )
  - 행과 열의 크기 확인

```
x <- matrix(c(1:9), ncol=3)
x
t(x)

ncol(x)
nrow(x)
```

# Element selection (1/4)

- Vector
  - By index

```
LETTERS[c(1, 5, 6)]
LETTERS[-c(1, 5, 6)]
rev(LETTERS)
```

  - By name

```
aaa <- 1:10
names(aaa) <- letters[1:5]
names(aaa)
aaa[c('a', 'c', 'e')]
```

  - By logical

```
aaa <- 1:5
aaa[c(T, F, F, T, T)]
```

  - Recycle

```
aaa[c(T, F)]
aaa[c(F, T)]
```

  - NA

```
aaa <- LETTERS
aaa[50]
aaa <- 1:10
names(aaa) <- LETTERS[1:10]
aaa['Z']
```

# Element selection (2/4)

- Array

```
aaa <- matrix(1:16, 4, 4)
aaa
aaa[c(1,3), c(2,4)]
aaa[5]

dimnames(aaa) <- LETTERS[1:4]
dimnames(aaa)[[1]] <- LETTERS[1:4]
aaa

dimnames(aaa)[[2]] <- letters[1:4]
aaa
row.names(aaa) <- LETTERS[1:4]
colnames(aaa) <- letters[1:4]

aaa[c('A', 'C'), c('a', 'd')]
aaa[1:2, ]
```

# Element selection (3/4)

- List
  - [ selects sub-lists
  - [[ selects an element within a list

```
list.ex <- list(a = c(1, 2, 3), b = c('a', 'b', 'c'), c = list(var1 = 'a', var2 = 'b'))
class(list.ex[2])
class(list.ex[[2]])
list.ex[['b']]
list.ex[[1:3]]
list.ex$a
```

# Element selection (4/4)

- Data frame

```
test.data.frame <- data.frame(Var1 = 1:10, Var2 = LETTERS[1:10])
test.data.frame$Var1
test.data.frame[['Var1']]
test.data.frame[[1]]

test.data.frame[5, 1]
test.data.frame[5, 'Var1']
test.data.frame[5, c(T, F)]

subset(test.data.frame, Var1 >= 8)
test.data.frame <- data.frame(Var1 = 1:10, Var2 = LETTERS[1:10], Var3 = LETTERS[11:20])
subset(test.data.frame, Var1 >= 8, select = c(Var1, Var3))
subset(test.data.frame, Var1 >= 8, select = -Var2)

idx <- which(test.data.frame$Var1 >= 8)
test.data.frame[idx, ]
```

# Control Structure (1/2)

- if … else

```
a <- 5
if (a > 0) {print ('a is greater than 0')} else
{print ('a is smaller than 0')}

a <- 10
if (a < 0) {
  print ('a is smaller than 0')} else if (a >= 0 & a <= 5)
  {print ('a is between 0 and 5')} else
  {print ('a is greater than 10')}

if (a < 0) {print ('a')}
else {print ('b')}

if (a < 0) {print ('a')
  } else {print ('b')}
```

- while

```
a <- 1
while (a < 4) {
  print (paste ('This is iteration', a))
  a <- a + 1
}
```

# Control Structure (2/2)

- for

```
vector <- c('aaa', 'bbb', 'ccc')
for (i in vector) {
  print(i)
}

num <- c(1:3)
for (i in num) {
  print (i + 1)
}
```

- switch

```
inp <- 'b'
switch(inp,
       a = print ('inp is a'),
       b = print ('inp is b'),
       c = print ('inp is c'),
       print ('inp is not a, b, c'))
inp <- 1
switch(inp,
       1 = print (inp + 1))
inp <- 1
switch(inp,
       '1' = print (inp + 1),
       print ('none'))
inp <- 2
switch(inp, inp+1, inp+2, inp+3, inp+4)
```

# Reading data (1/4)

- Delimited data
  - header: 첫 열이 제목
  - nrows: 읽을 열의 개수
  - skip: 무시 할 열의 개수
  - encoding: 문자 코드 (한글)

- Line by line

```
data <- readLines(path)
class(data)
length(data)
```

- Character set

```
data <- readChar(path, nchars = 1e5)
class(data)
length(data)
```

```
path <- 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
data <- read.table(path, sep=',')
class(data)
```

# Reading data (2/4)

- JSON
  - JavaScript Object Notation
  - RJSONIO

```
library(RJSONIO)
url <- 'http://api.worldbank.org/v2/datacatalog?format=json'
json <- fromJSON(url)
```

  - rjson

```
library(rjson)
raw.json <- readChar(url, nchars = 1e6)
json <- fromJSON(raw.json)
```

- XML

```
library(XML)
url <- 'http://api.worldbank.org/v2/datacatalog?format=xml'
xml.obj <- xmlTreeParse(url)
class(xml.obj)
```

# Reading data (3/4)

- SQL
  - RJDBC
  - RODBC
  - DBI
    - RMySQL
    - ROracle
- External source
  - Excel
    - xlsx
    - openxlsx
  - SAS, SPSS
    - Hmisc
    - Foreign
- BEST → .csv

# Reading data (4/4)

- Excel

```
library(openxlsx)

## 엑셀 파일 입력
xlsxFile <- " https://github.com/awalker89/openxlsx/raw/master/inst/readTest.xlsx"
xlsx.data <- read.xlsx(xlsxFile)
View(xlsx.data)

## 엑셀 파일 출력
## 오류 : write.xlsx(data.csv, file='iris.R.xlsx')
## Rtools 필요
```

# Contents

# View

- Size and Structure

```
dim(iris)
names(iris)
str(iris)
```

- Attributes of Data

```
attributes(iris)
```

# Explore

- Summary

  summary(iris)

- Mean, Median, Range, Quantile

  range(iris$Sepal.Length)
  quantile(iris$Sepal.Length)
  quantile(iris$Sepal.Length, c(0.1, 0.3, 0.65))

- Variance, Histogram

  var(iris$Sepal.Length)
  hist(iris$Sepal.Length)

# sorting

- sort()

```
vec1 <- c(3, 2, 5, 1, 4)
sort(vec1)
sort(c(T, T, F, F))
sort(c('play', 'plan', 'plot', 'proof'))
```

- order()

```
vec1 <- c(3, 2, 5, 1, 4)
order(vec1)
```

- sort() vs order()

```
vec1[order(vec1)]
vec1 <- c(3, 2, 5, 1, 4)
vec2 <- c(2, 2, 3, 3, 1)
order(vec2, vec1, decreasing = c(T, F))
order(vec2, vec1, decreasing = c(F, T))

data('iris')
names(iris)
iris.ordered <- iris[order(iris$Sepal.Length,
iris$Sepal.Width), ]
```

# apply function

- lapply(), vapply(), sapply(), apply()

```
sample.list <- list(a = runif(100, 0, 1), b = runif(500, 0, 100), c = runif(35, 0, 200))
sapply(sample.list, quantile, probs = 0.75)
sapply(sample.list, function(x) round(sum(x + 2)))
sapply(sample.list, function(x) quantile(x, probs = 0.75))
vapply(sample.list, quantile, probs = 0.75, c('Mean' = 0))
lapply(sample.list, quantile, probs = 0.75)

x <- as.numeric(c(1:2000000))
ptm <- proc.time()
for (i in x) x[i] <- x[i] * x[i]
proc.time() - ptm
ptm <- proc.time()
x <- sapply(x, function(x) x * x)
proc.time() - ptm
```

# 데이터 결합 (1/3)

- rbind()

```
rbind(c(1, 2, 3), c(4, 5, 6))
rbind(c(1, 2, 3), c(4, 5))    # 오류
rbind(c(1, 2, 3), c(4, 5, ))  # 오류
rbind(c(1, 2, 3), c(4, 5, NA))

x <- data.frame(id = c(1, 2), name = c('a', 'b'))
x
y <- rbind(x, c(3, 'c'))  ## 오류 요인 수준 다름
y <- rbind(x, c(3, 'b'))
y

x <- data.frame(id = c(1, 2), name = c('a', 'b'), stringsAsFactors = F)
y <- rbind(x, c(3, 'c'))
y
```

# 데이터 결합 (2/3)

- cbind()

```
cbind(c(1,2,3), c(4,5,6))

x <- data.frame(id = c(1, 2), name = c('a', 'b'), stringsAsFactors = F)
y <- cbind(x, greek = c('alpha', 'beta'))
y
summary(y)
y <- cbind(x, greek = c('alpha', 'beta'), stringsAsFactors=F)
y
summary(y)

y$korean <- c('ㄱ', 'ㄴ')
y
summary(y)
```

# 데이터 결합 (3/3)

- merge()

```
x <- data.frame(names=c('a','b','c'), math=c(40,50,70))
y <- data.frame(names=c('a','b','c'), english=c(60,40,80))
merge(x, y)
cbind(x, y)

y <- data.frame(names=c('a','b','c', 'd'), english=c(60,40,80,90))
merge(x, y)
merge(x, y, all=T)
```

# 데이터 분리

- split(), subset()

```
idx <- sample(2, nrow(iris), replace = T, prob = c(0.3, 0.7))
idx
split(iris[idx==1,], iris[idx==1,]$Species)

subset(iris, Species == 'setosa')
subset(iris, Species == 'setosa' & Sepal.Length > 5.1)
subset(iris, select = c(Sepal.Length, Species), Species == 'setosa' & Sepal.Length > 5.1)
subset(iris, select = -c(Sepal.Length, Species), Species == 'setosa' & Sepal.Length > 5.1)
```

# which

- which(), which.min() , which.max()

```
iris$Sepal.Length > 6.0
which(iris$Sepal.Length > 6.0)
iris[which(iris$Sepal.Length > 6.0),]

which.min(iris$Sepal.Length)
iris[which.min(iris$Sepal.Length),]

which.max(iris$Sepal.Length)
iris[which.max(iris$Sepal.Length),]
```

# dplyr (1/4)

- 조건에 맞는 데이터만 추출

```
library(dplyr)

data('iris')
iris %>%  filter(Species == "setosa")
iris %>%  filter(Sepal.Length > 7)
iris %>%  filter(Sepal.Length > 6) %>% filter(Sepal.Width > 3.5)
iris %>%  filter(Sepal.Length > 6 & Sepal.Width > 3.5)
iris %>%  filter(Sepal.Length > 6 | Sepal.Width > 3.5)

iris %>% filter(Species == "setosa" | Species == "virginica")
iris %>% filter(Species %in% c("setosa", "virginica"))
```

%>% : Ctrl + Shift + M

# dplyr (2/4)

- 필요한 변수만 추출

```
iris %>% select(Species)
iris %>% select(Sepal.Length, Petal.Length, Species)

iris %>% select(-Species)
iris %>% select(-Sepal.Length, -Sepal.Width)

iris %>% select(Sepal.Length, Species) %>%
  filter(Species == "setosa" & Sepal.Length > 5.4)

iris %>% select(Sepal.Length, Species) %>%
  filter(Species == "setosa" & Sepal.Length > 4) %>%
  head
```

# dplyr (3/4)

- 정렬하기

```
iris %>% arrange(Sepal.Length)
iris %>% arrange(desc(Sepal.Length))
iris %>% arrange(Sepal.Length, Sepal.Width)
```

- 파생변수 추가하기

```
iris %>%
  mutate(Sepal.total = Sepal.Length + Sepal.Width) %>% head

iris %>%
  mutate(Sepal.total = Sepal.Length + Sepal.Width,
         Petal.mean = (Petal.Length + Petal.Width)/2) %>% head

iris %>%
  mutate(Size = ifelse(Sepal.Length >= 6, "big", "small")) %>% head
```

# dplyr (4/4)

- 요약하기

```
iris %>% summarise(Sepal.Length.mean = mean(Sepal.Length))

iris %>% group_by(Species) %>% summarise(Sepal.Length.mean = mean(Sepal.Length))

iris %>% group_by(Species) %>%
  summarise(Sepal.Length.mean = mean(Sepal.Length),
            Sepal.Length.sum = sum(Sepal.Length),
            Sepal.Length.median = median(Sepal.Length),
            Sepal.Length.max = max(Sepal.Length),
            n = n())

mpg <- ggplot2::mpg
View(mpg)
mpg %>% group_by(manufacturer, drv) %>%
  summarise(mean_cty = mean(cty)) %>% head
```

# 그룹별 연산

- table()
  - 두 요인의 조합별 개수 생성
  - 데이터 분석의 분류 결과 확인에 사용
  - table (객체,...)

```
table(by1, by2)
iris.Species.sample <- sample(iris$Species)
table(iris$Species, iris.Species.sample)
```

# Summary Chart

- Pie Chart

```
table(iris$Species)
pie(table(iris$Species))
```

- Bar Chart

```
barplot(table(iris$Species))
```

# Save to Files

- Charts

```
pdf('myPlot.pdf')
x <- 1:50
plot(x, log(x))
graphics.off()
```

- Text

```
write.csv(iris, file = 'iris.txt')
write.csv(iris, 'iris.data.csv', row.names = F)
```

- RData

```
save(iris, file = 'iris.RData')
```

## 2. 현장사례

| 국가명 | 회사명 | 주제 | ROI |
|---|---|---|---|
| 미국 | First Union | 고객 만족도 향상 mass 마케팅에서 일대일 마케팅으로의 전환 | 캠패인 반응 결과 60% 향상 |
| 미국 | Bank of America | 20%의 은행 고객이 150%의 이익에 공헌을 하며, 40-50%에 해당하는 고객이 은행 전체의 50%를 감소시킨다. | 50%의 이익을 감소 시키는 고객 중 상위 20% 고객을 탐지 |
| 미국 | Wells Fargo | 기존의 모형에 대한 불만족 해소 | 기존의 주 단위의 예측 모형에서 일단위로의 전환을 통해 보다 신속한 파악이 가능해짐 |
| 독일 | Deutsche Bank | Credi scoring 수익성 높은 고객의 행동 예측 | 보다 정밀한 모형을 찾게 되었으며, 이에 소비되는 시간 감축 |
| 캐나다 | Toronto Dominion Bank | 다양한 모델링 기법과 실행에 있어서의 신속함을 요구 | 10주로 계획이 되었던 프로젝트가 3-4주로 단축이 됨 |
| 미국 | Newport News | 제한된 카탈로그 분석을 통한 고객 세분화 | 30 시간이 걸리던 분석 시간이 30 초로 단축이 됨 |
| 미국 | AT&T | 모델링 프로세스의 비효율성 개선 | 모형 적합과정에서의 50% 시간 단축 |

## 3. 예제 또는 실습

- R, RStudio 설치
- vector 데이터 생성 및 matrix, data.frame 데이터로 변환
- 부분 matrix, data.frame 생성
- 데이터 요약 정보 확인
- 데이터 형태 변환
- mpg 데이터로 회사별로 “compact” (경차)의 도시 연비 (cty), 고속 도로 연비 (hwy)의 통합 연비의 평균을 구해 성능이 좋은 차 1 ~ 5위 까지 출력하기

## 4. 학습진단 평가문제

- Data Mining의 주요 Keyword가 아닌 것은?

  ① Automated　② Hidden　③ Retrospective　④ Predictive

- Data Mining과 연관 분야가 아닌 것은?

  ① Statistics　② Event Processing

  ③ Machine Learning　④ Optimization

- Decision Tree, Regression Model, Rule Induction, Neural Network 모델을 사용자가 이해하기 쉬운 순서대로 나열한 것은?

  ① Decision Tree – Rule Induction – Regression Model – Neural Network

  ② Rule Induction – Regression Model – Neural Network – Decision Tree

  ③ Decision Tree – Regression Model – Rule Induction – Neural Network

  ④ Rule Induction – Decision Tree –Regression Model – Neural Network

## 4. 학습진단 평가문제

- Data Mining Application으로 알맞지 않은 것은?

  ① Market analysis ② Fraud detection ③ Face detection ④ Stream mining

- 데이터에 대한 초기 이해를 위해 사용하는 방법은?

  ① Supervised Learning ② Classification

  ③ Forecasting ④ Unsupervised Learning

- KDD 단계 중 가장 많은 노력이 필요한 단계는?

  ① Objective Determination ② Data Preparation

  ③ Data Mining ④ Evaluation

- R의 특징으로 알맞지 않은 것은?

  ① In-Disk ② OOP ③ Package ④ OSS

## 4. 학습진단 평가문제

- R의 변수의 특징으로 알맞지 않은 것은?

① overriding　② 대소문자 구분

③ overfitting　④ 타입 선언 불필요

- R의 atomic class가 아닌 것은?

① Character　② Numeric　③ Logical　④ String

- 2 dimension을 가지고, 모든 데이터 속성이 동일한 class는?

① vector　② list　③ matrix　④ data.frame

## 3. 예제 정답

```
mpg %>%
  filter(class == 'compact') %>%
  group_by(manufacturer) %>%
  mutate(toty = (cty + hwy)/2) %>%
  summarise(toty.mean = mean(toty)) %>%
  arrange(desc(toty.mean)) %>%
  head(5)
```