Data Mining with R

1. 데이터 마이닝과 R

Contents

문제

- 당신은 중개 회사의 마케팅 매니저이다.
  - 문제 : Churn (이탈 고객)이 너무 많음
    - 40% Turnover (6개월의 입문 기간이 끝나고)
  - 고객은 계좌가 개설되면 인센티브를 받는다.
  - 떠나려고 하는 모든 고객에게 인센티브를 주는 것은 비용이 너무 많이 든다.
  - 떠난 고객을 다시 데려오는 것도 어렵고, 비용이 많이 드는 일이다.

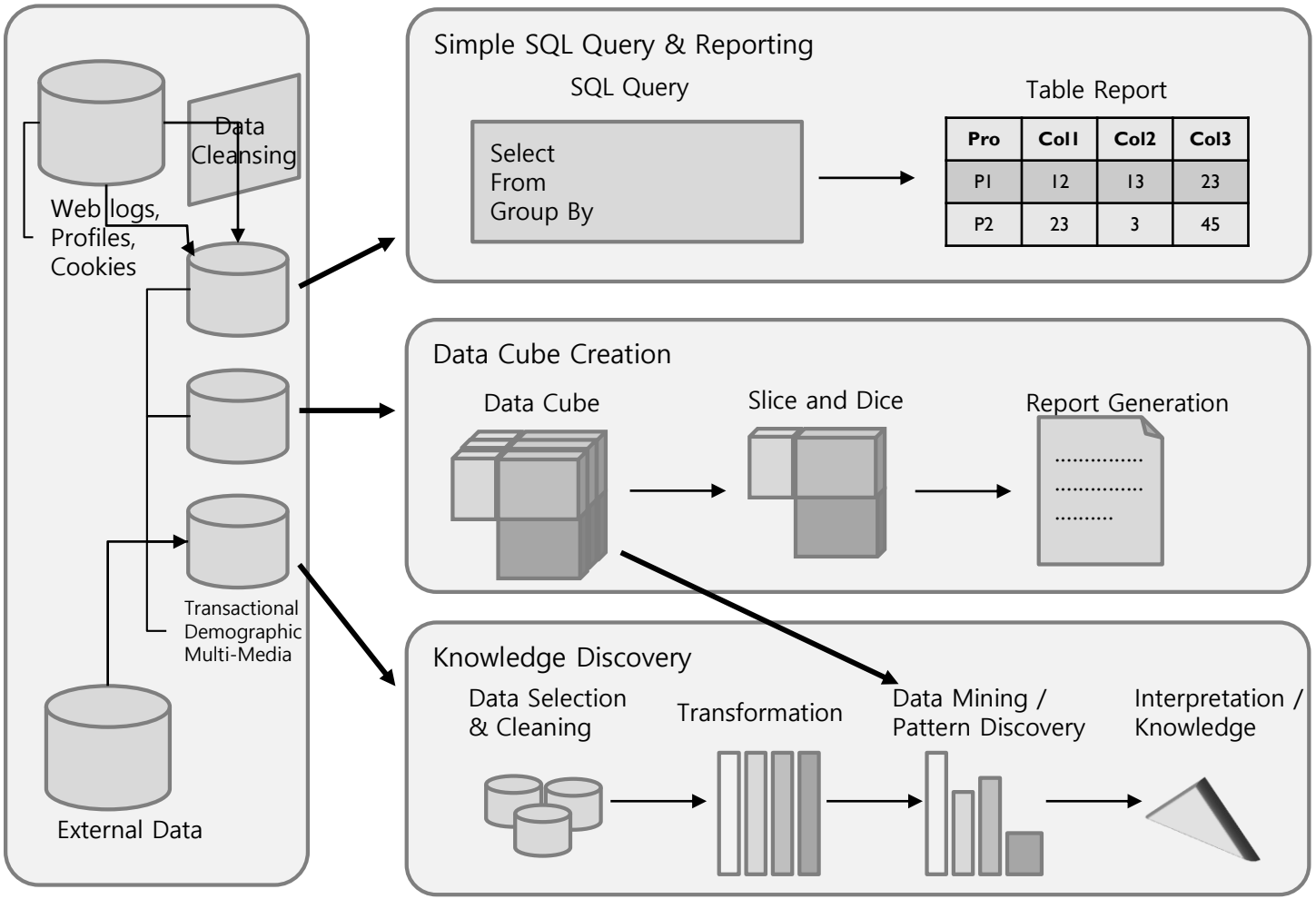
해법

- 입문 기간이 끝나기 1개월 전에 떠날 고객 예측
  - 이탈로 예측된 고객을 유지하기 위해서, 예측에 근거한 무언가를 제시해야 함
    - 이탈로 예측되지 않은 고객은 고려하지 않음
  - 고객을 유지하고 싶지 않으면, 아무 것도 안 함
- 미래의 행위를 어떻게 예측할 것인가?
  - Tarot Cards
  - Magic 8 Ball
  - 관상 / 사주 팔자

The Big Picture

- Data Mining에 대한 잘못된 가정과 선입관이 많음
- Data Mining은 훨씬 큰 프로세스의 일부분 임
  - 10% of 10% of 10% of 10%
  - 정확도가 가장 중요한 측정치인 것은 아님
- 데이터 자체가 중요함
- 알고리즘은 생각만큼 중요하지는 않음
- Data Mining으로 발견한 패턴을 이해하지 못하면, 영향을 끼칠 수 없음 (다른 사람을 납득시킬 수 없음)

Computational Knowledge Discovery



용어

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

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

Data Mining의 정의

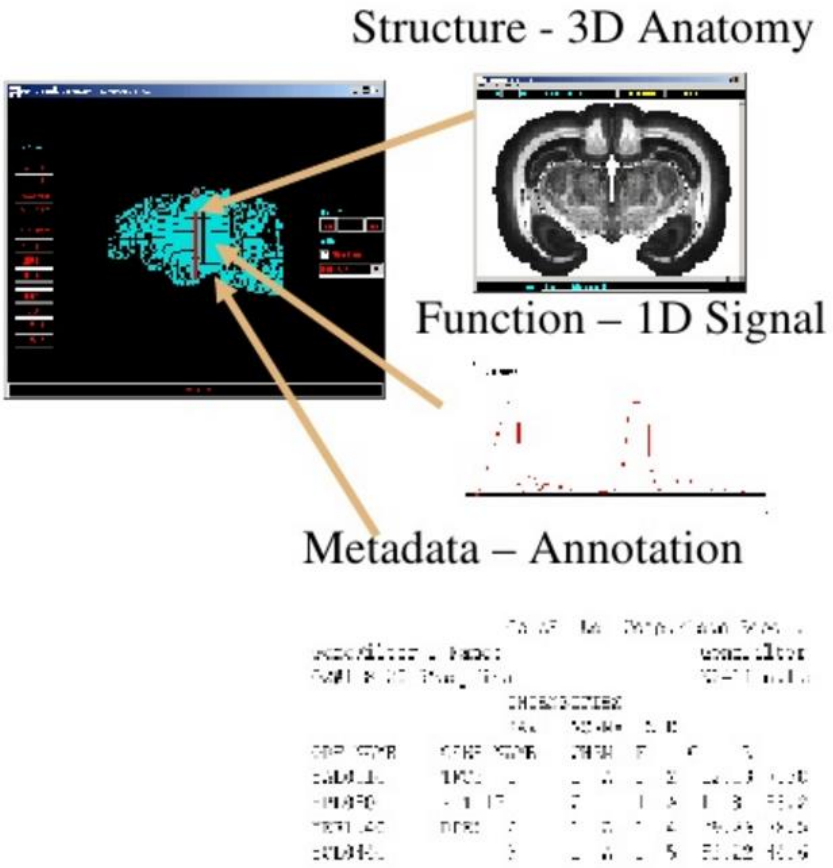
- The automated extraction of hidden predictive information from (large) databases
- Three key words:
  - Automated
  - Hidden
  - Predictive
- Data Mining은 상황을 주도하게 함
  - Retrospective (회고)가 아닌 Prospective (미래의)

Data Mining의 목적

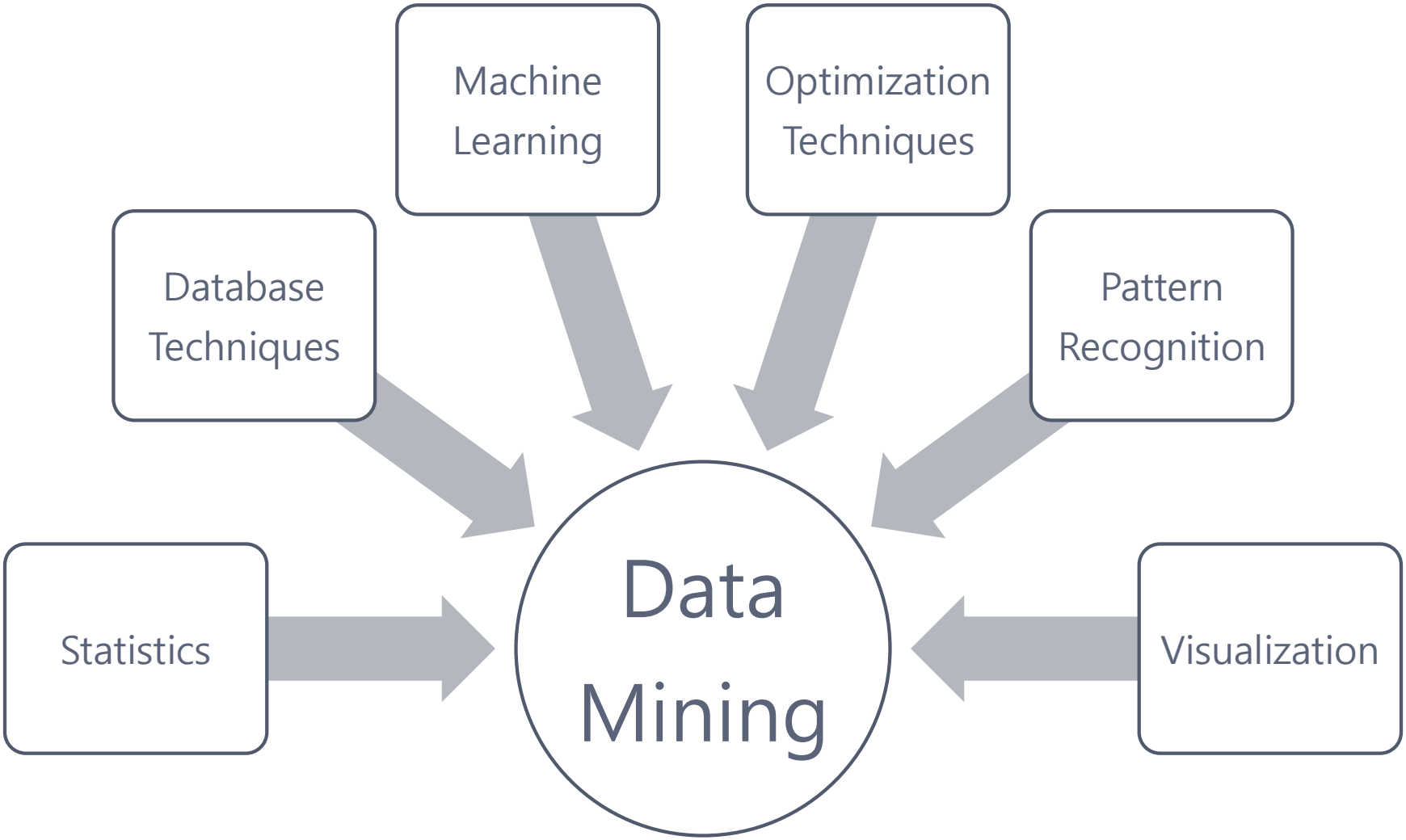
- 데이터 소스에서 모델 생성까지의 전반적인 통계적인 과정을 단순화하고 자동화하는 것
- Many different data mining algorithms / tools
- Statistical expertise required to compare different techniques
- Build intelligence into the software

데이터의 종류

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



연관 분야


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

필요성

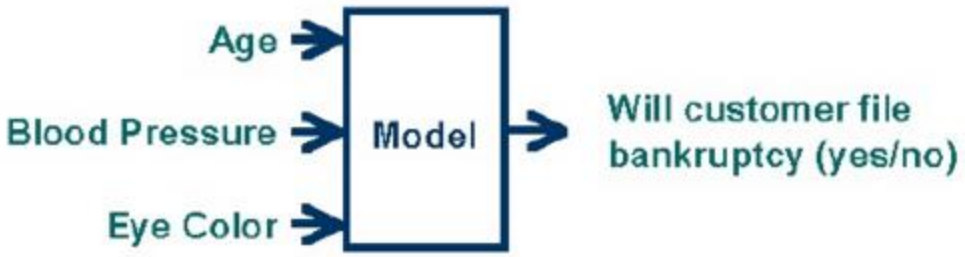
- 전통적인 분석 방법론을 사용하기에는 데이터가 너무 크다.
  - 데이터 개수의 증가 (10^9 – 10^{12} Bytes -> GB - TB)
  - 다차원 데이터 (100 - 1000 속성)
- 기업의 데이터 자산
  - 소규모(5-10 %) 데이터 만 분석
  - 분석할 수 없는 데이터도 저장
- 데이터베이스가 확대되면서 전통적인 query language에 의한 의사 결정이 어려워 짐
  - Query language에 의해서 다양한 관점을 분석하기 어려움

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

정의: Predictive Model (예측 모델)

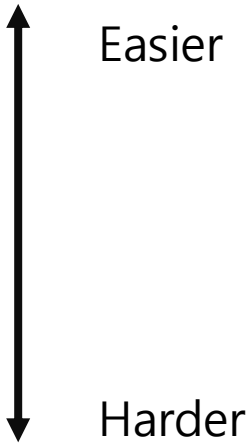
- 과거와 현재의 정보를 기반으로 미래를 예측하는 'black box'



- 다량의 입력이 필요함

Models

- 몇몇 모델은 다른 것보다 우수함
  - 정확도
  - 이해도
- 모델의 이해도
  - Decision trees
  - Rule induction
  - Regression models
  - Neural Networks



Scoring

- The workhorse of data mining
- 모델은 한 번 만들어지고, 지속적을 사용됨
- Data mining 결과를 사용하는 사람은 data mining model을 만드는 사람과는 다름
  - Model 전달하는 방법에 대한 고민 필요
- Issue: model 개발 시 사용된 데이터와 model을 사용할 대 사용되는 데이터의 조화
  - 데이터가 동일한가?
  - 일관성이 자동적으로 유지되는가?

Model 사용 방법

- Qualitative (질적)
  - 다루고 있는 데이터에 대한 insight 제공
    - If city = New York and 30 < age < 35 …
    - 중요한 연령대는 20에서 25
    - 서면 캠페인 대상을 New Yorker로 변경
  - 상호 작용 및 좋은 시각화가 필요
- Quantitative (양적)
  - 자동화 과정
  - 매일 자정에 새로운 유전자 chip 데이터 집합의 에러 Score 계산
  - 결론에 기반 (Bottom-line orientation)

Data Mining Applications

- Market analysis
- Risk analysis and management
- Fraud detection and detection of unusual patterns (outliers)
- Text mining (news group, email, documents) and Web mining
- Stream data mining
- DNA and bio-data analysis

Market Analysis

- 데이터 소스
  - 신용 카드 transactions, loyalty cards, discount coupons, customer complaint calls
- Target marketing
  - 동일 특성을 가지는 고객들의 군집 모델 생성 – 관심, 수입, 소비 형태 등
  - 지속적으로 고객의 구매 패턴 분석
- Cross-market analysis
  - 제품 판매 사이의 관계 분석을 통한 예측 수행
- Customer profiling
  - 어떤 성향의 고객이 어떤 제품을 구매할 지 분석
- Customer requirement analysis
  - 다양한 고객 사이의 최적의 제품 선정
  - 새로운 고객에 영향을 끼치는 요소 예측

Corporate Analysis & Risk Management

- Finance planning and asset evaluation
  - 현금 흐름 분석과 예측
  - Time series analysis (financial-ratio, trend analysis 등)
- Resource planning
  - 자원과 지출 비교 및 요약
- Competition
  - 경쟁사와 시장 방향 모니터링
  - 고객 군집화 및 군집 기반 가격 정책 수립
  - Red ocean 시장에서의 가격 정책 수립

Fraud Detection & Mining Unusual Patterns

- 적용 영역 : Health care, retail, credit card, telecomm.
  - 돈 세탁 : 의심되는 통화 흐름 추적
  - Medical insurance
    - Professional patients, ring of doctors
    - Unnecessary or correlated screening tests
  - Telecommunications : phone-call fraud
    - Phone call model : destination of the call, duration, time of day of week. Analyze patterns that deviate from an expected norm
  - Retail industry
    - Analysis estimate that 38% of retail shrink is due to dishonest employees
  - Anti-terrorism

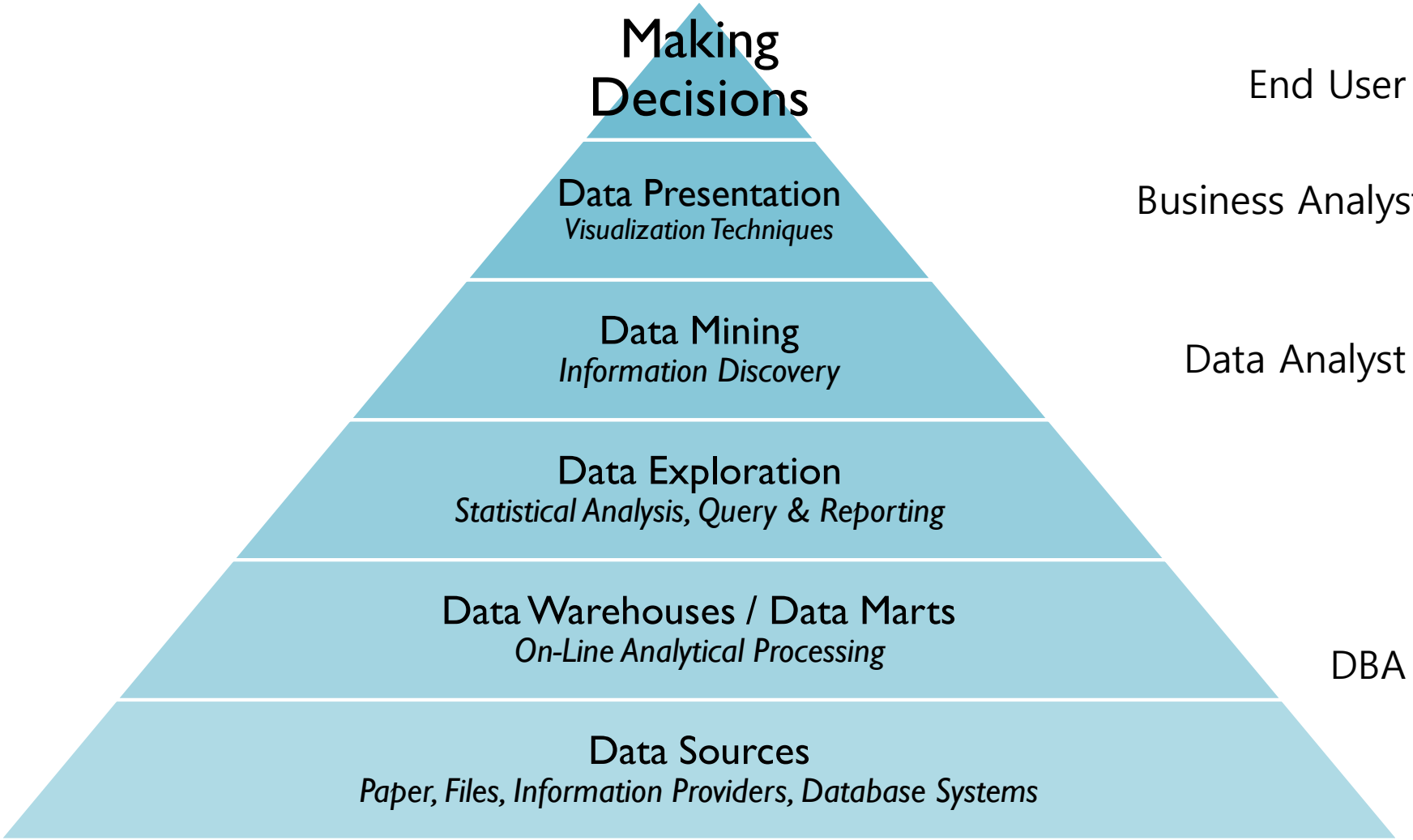
Data Mining 문제의 종류

- Classification / Segmentation (분류)
  - Binary (Yes/No)
  - Multiple category (Large/Medium/Small)
- Forecasting
- Association rule extraction
- Sequence detection
  - Gasoline Purchase → Jewelry Purchase → Fraud
- Clustering (군집화)

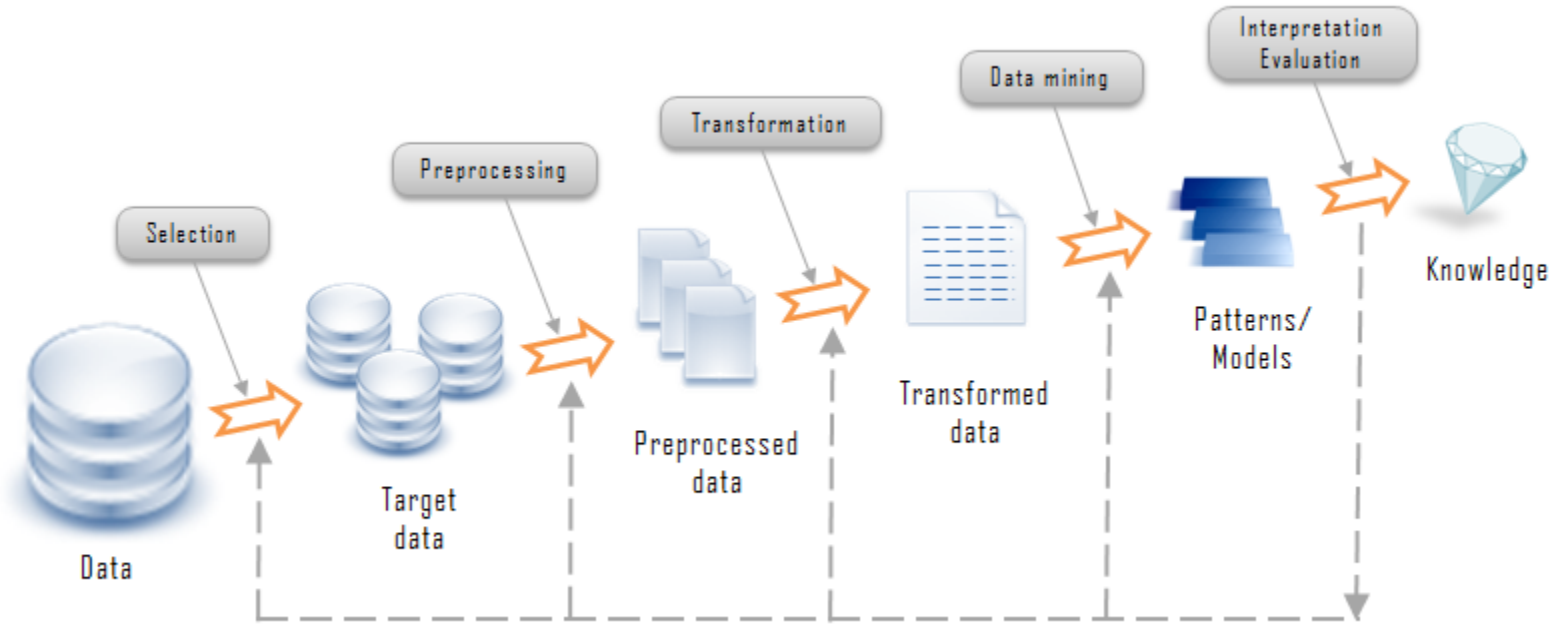
Supervised vs. Unsupervised Learning

- Supervised: Problem solving, 교사
  - 실 비즈니스 문제와 과거 데이터에 기반
  - 결과의 품질은 데이터의 품질에 관련
- Unsupervised: Exploration (clustering), 비 교사
  - 데이터에 대한 초기 이해에 유용
  - 불분명한 패턴이 나타나기도 함

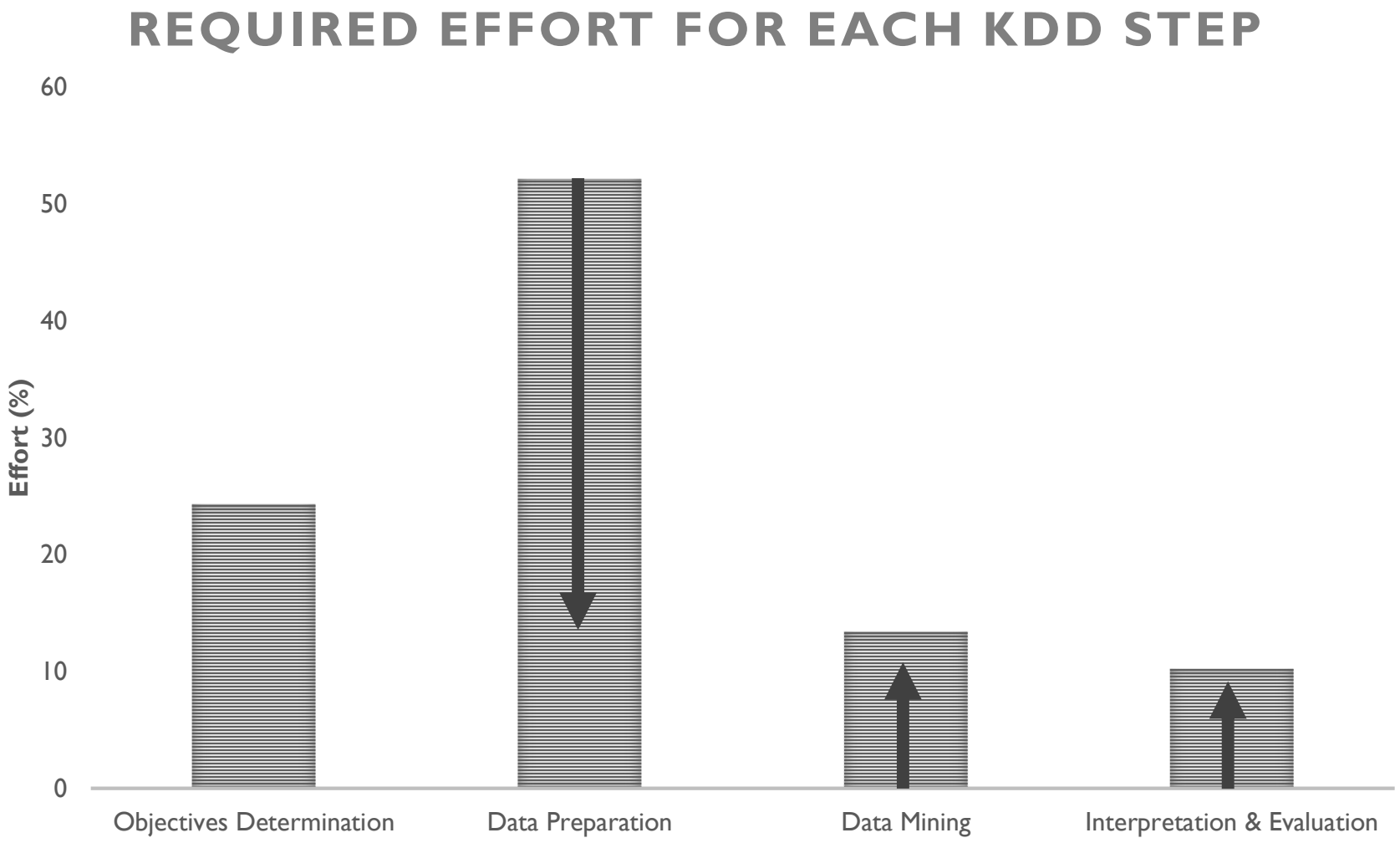
Data Mining and Business Intelligence



Knowledge Discovery



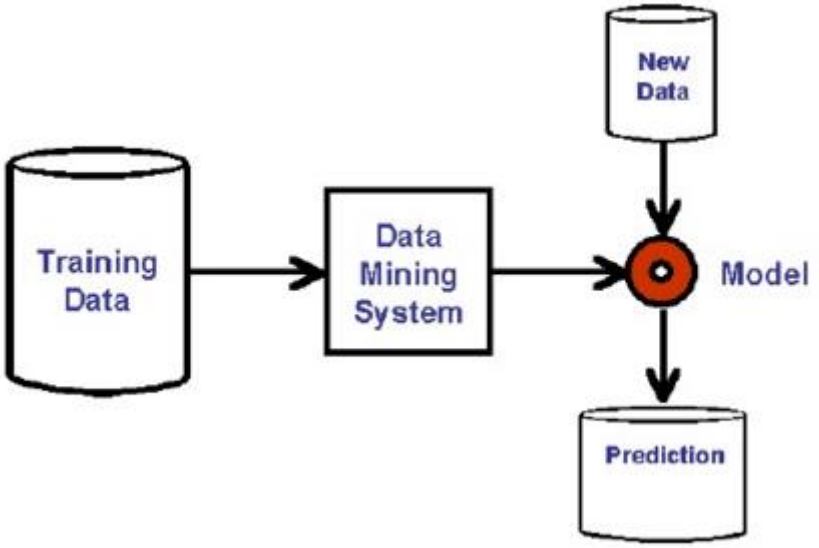
Required effort for each KDD Step



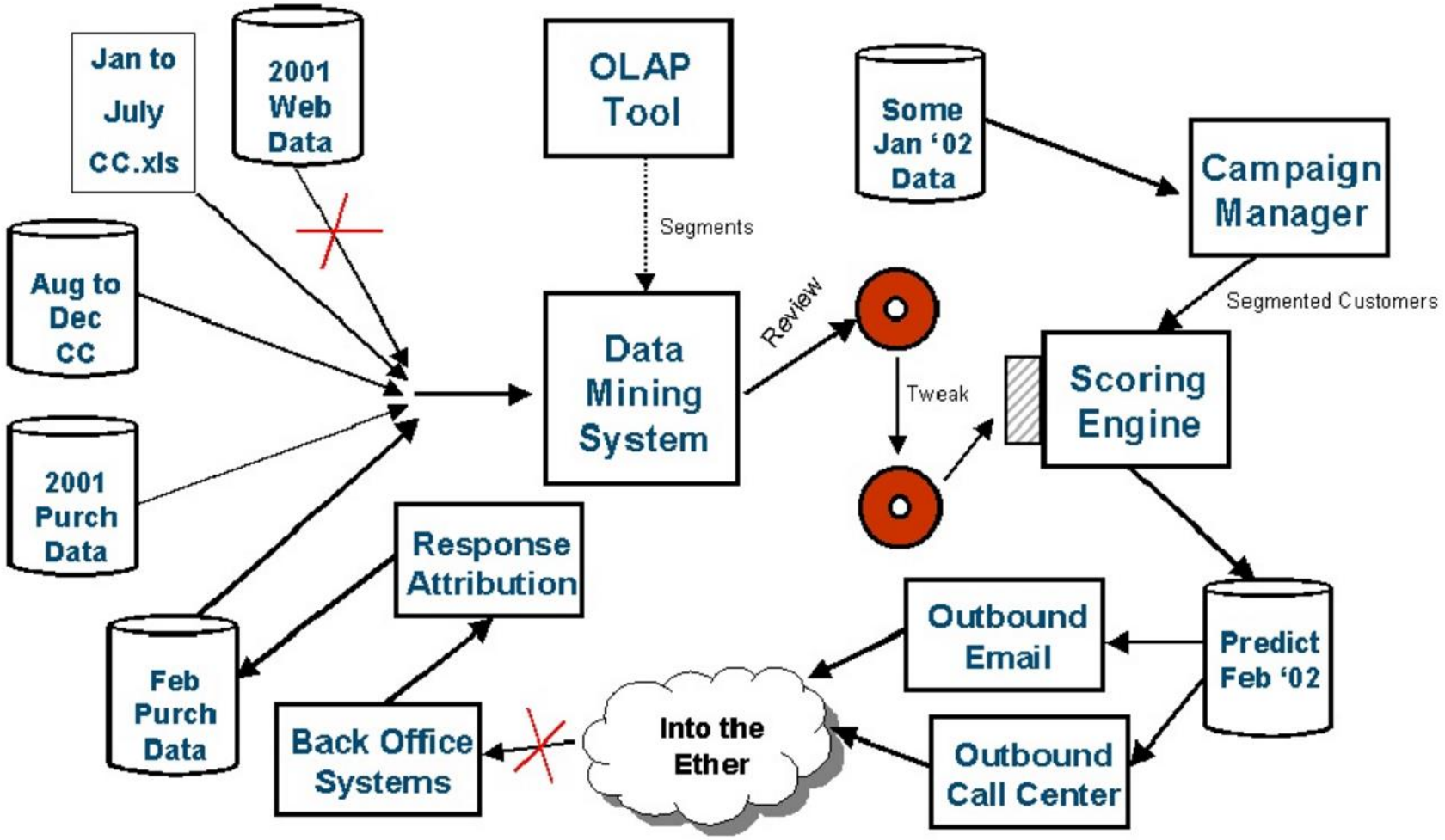
CRISP-DM


Business Understanding
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Data

How are Models Built and Used?



What the Real World Looks Like



What Caused this Complexity?

- Volume
  - Much more data
    - More detailed data
    - External data sources
  - Many more data segments
- Speed
  - Data flowing much faster
  - Errors can be easily introduced into the system
- Desire to include business inputs to the process
  - Financial constraints

Legal and Ethical Issues

- Privacy Concerns
  - Becoming more important
  - 데이터가 사용되고 분석되는 방법에 영향을 끼침
  - 소유권 이슈
- Government regulation of particular industry segments
  - FDA rules on data integrity and traceability
- Often data included in a data warehouse cannot legally be used in decision making process
  - Race, Gender, Age
- Data contamination will be critical

Contents

R

- S Language : 1976년 Bell Lab의 John Chambers, Rick Becker, Allan Wilks에 의해 개발된 데이터 분석 언어
- S-plus : 1988년에 Statistical Sciences, Inc. 에서 Bell Labs으로부터 License 받아 개발한 상용 솔루션
- 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)

- R is a language and environment for statistical computing and graphics.
  - 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
- R is available as Free Software under the terms of the Free Software Foundations GNU General Public License in source code form.
  - "GNU" 라는 이름은 "GNU's Not Unix!" 라는 문장의 약어
  - GNU는 자유 소프트웨어를 뜻 함
    - 프로그램을 어떠한 목적으로도 실행할 수 있는 자유
    - 자신의 필요에 맞게 개작 할 수 있는 자유
    - 복제물을 재 배포 할 수 있는 자유
    - 프로그램을 개선 시킬 수 있는 자유와 개선된 이점을 공동체 전체가 누릴 수 있게 발표할 자유

R이란? (2/2)

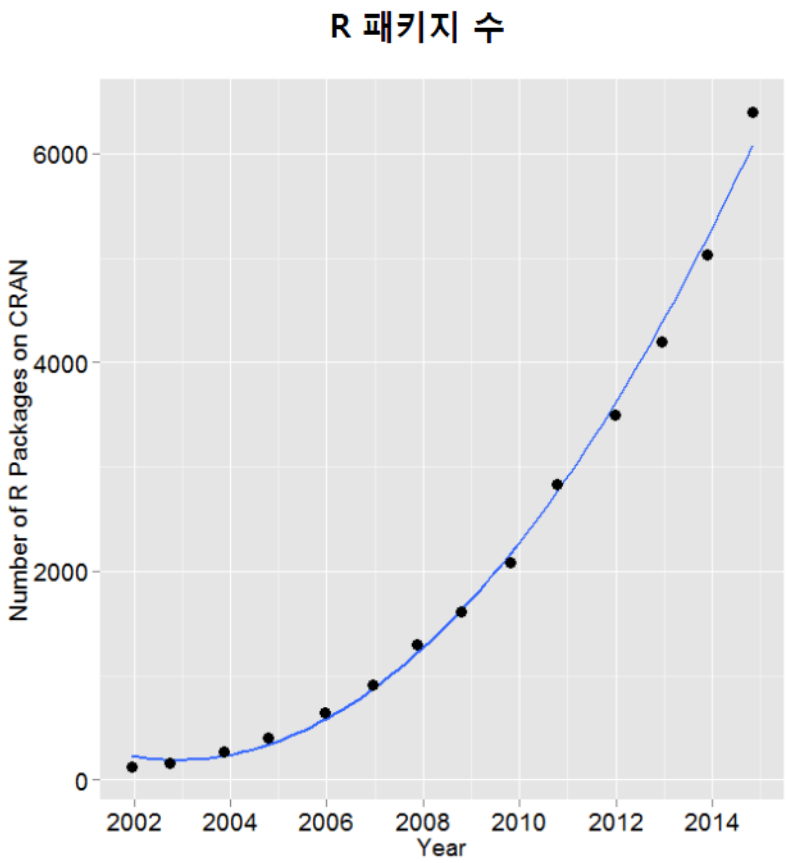
- R is Free.
  - 사용자의입장에서 Free 라는 것이 단지 무료임을 뜻하는 것은 아님
  - 언제 어디서든 다운로드 및 설치가 가능
  - Windows, Linux, Unix, Mac 등 다양한 운영체제에서 동작
  - 누구나 패키지(Package)를 만들어 다른 사람과 공유 가능
  - Java, Python, .Net, Visual Studio, C, C++ 등 다양한 개발 언어 및 플랫폼과 연동

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

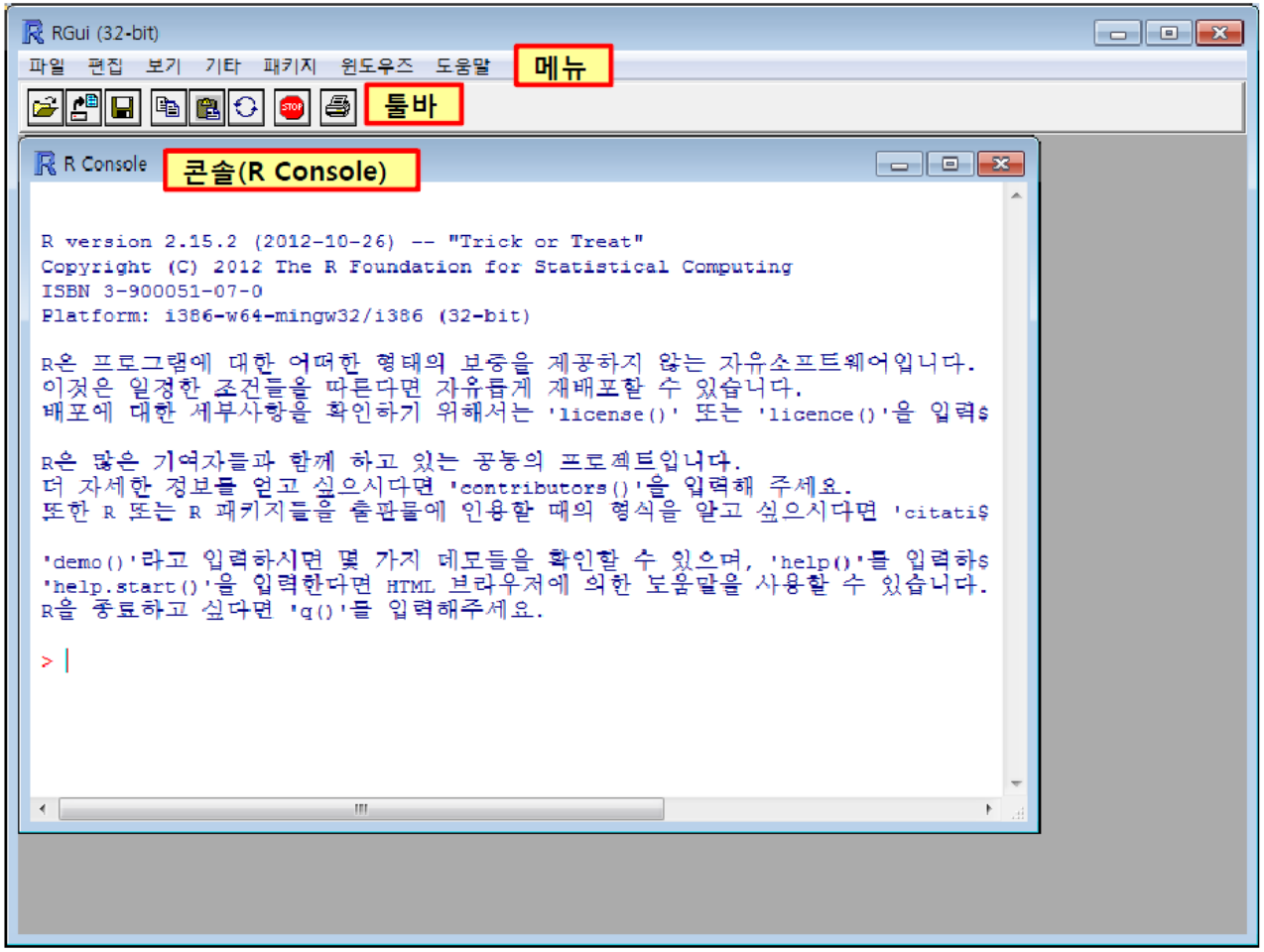
R Packages

- R Package (in CRAN)
  - CRAN (The Comprehensive R Archive Network)
  - CRAN Site에 7,909개 등록 됨 (2016년 2월 기준)
  - 새로운 통계 분석 알고리즘이나 새로운 IT 기술의 응용에 관한 것을 포함
    - ssizeRNA-Sample Size Calculation for RNA-SeqExperimental Design
    - mglmn-Model Averaging for Multivariate GLM with Null Models
  - 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"
```

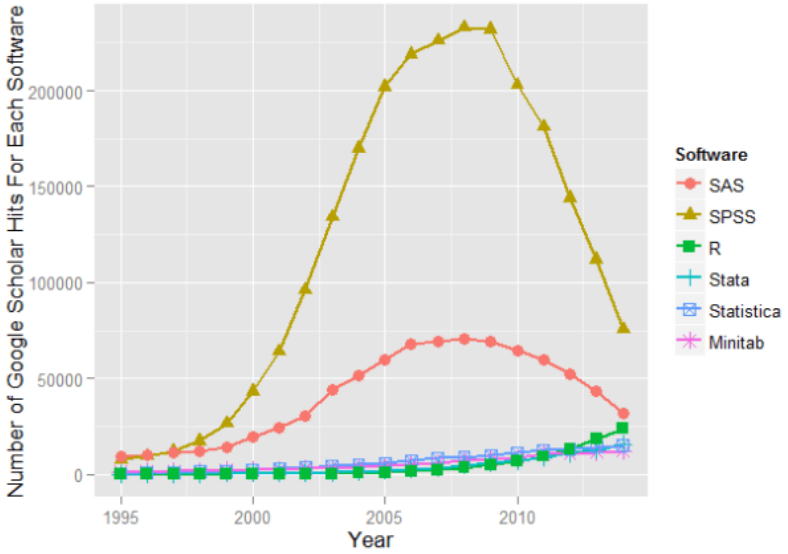
❑ **R Console에 입력된 R command**

- **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)

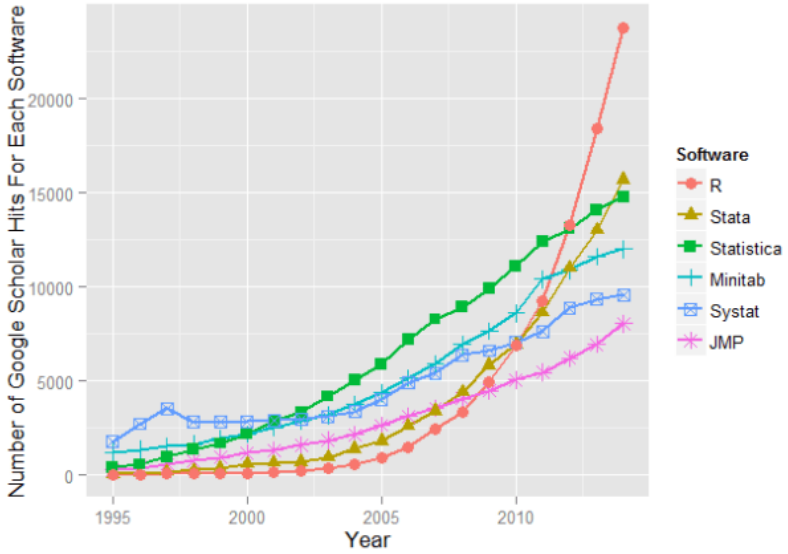
분석 패키지별 학술연구 활용 빈도(www.r4stats.com)

- SPSS, SAS는 둔화하는 추세임



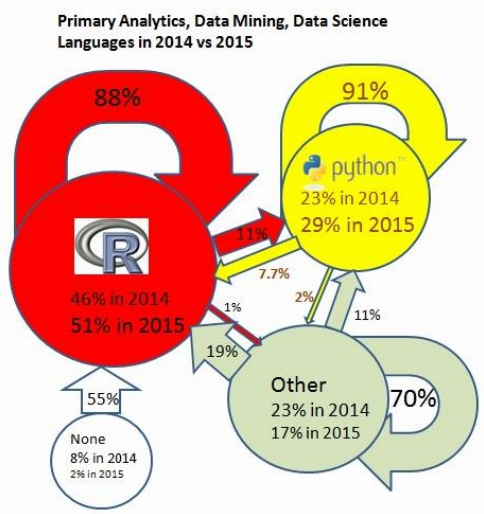
분석 패키지별 학술연구 활용 빈도 (상용 패키지 제외)

- 최근 R의 상승세가 매우 높으며, 2016년 말 상용 패키지 추월이 예상됨



Usage of R (2/2)

Your primary programming language for Analytics, Data Mining, Data Science tasks: [512 voters]	
2015 primary programming language:	
R (and its packages) (263)	51% (of 2015 votes)
Python (including scikit-learn and other libraries) (151)	29%
Other (Java, MATLAB, SAS, Scala, etc) (89)	17%
none (9)	1.8%
2014 primary programming language:	
R (and its packages) (237)	46% (of 2014 votes)
Python (including scikit-learn and other libraries) (117)	23%
Other (Java, MATLAB, SAS, Scala, etc) (118)	23%
none (40)	7.8%



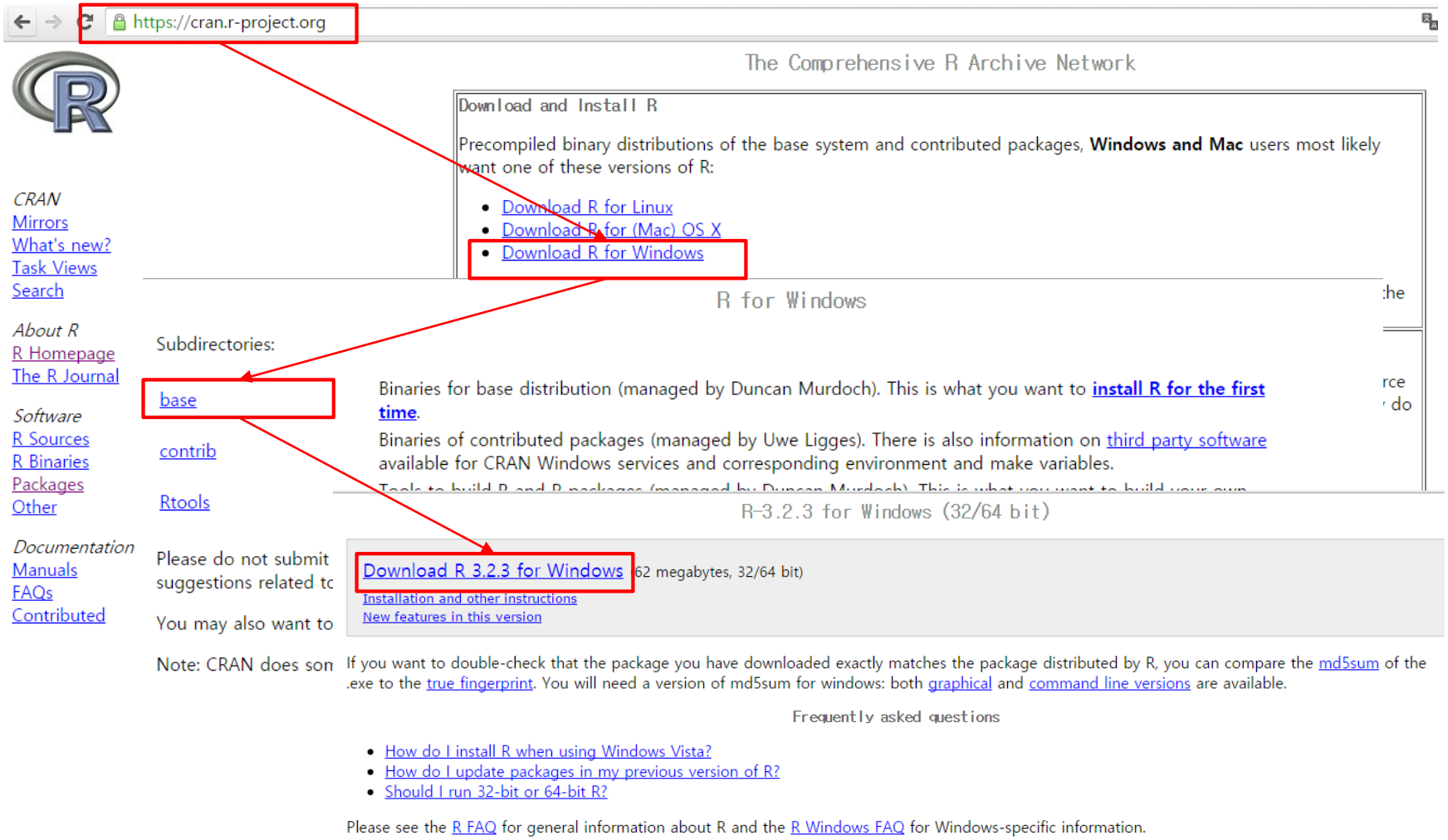
What Analytics, Big Data, Data mining, Data Science software you used in the past 12 months for a real project? [2759 voters]	
Legend: Red: Free/Open Source tools Green: Commercial tools Fuchsia: Hadoop/Big Data tools	% users in 2015 % users in 2014 % users in 2013
R (1293), 3.6% alone	46.9% 38.5% 37.4%
RapidMiner (870), 13.7% alone	31.5% 44.2% 39.2%
SQL (853), 0% alone	30.9% 25.3% na
Python (837), 0% alone	30.3% 19.5% 13.3%
Excel (631), 0% alone	22.9% 25.8% 28.0%
KNIME (553), 6.7% alone	20% 15.0% 5.9%
Hadoop (507), 0% alone	18.4% 12.7% 9.3%
Tableau (341), 0% alone	12.4% 9.1% 6.3%
SAS base (313), 0.6% alone	11.3% 10.9% 10.7%
Spark (311), 0% alone	11.3% 2.6% na
Weka (310), 0% alone	11.2% 17.0% 14.3%

출처: kdnuggets.com

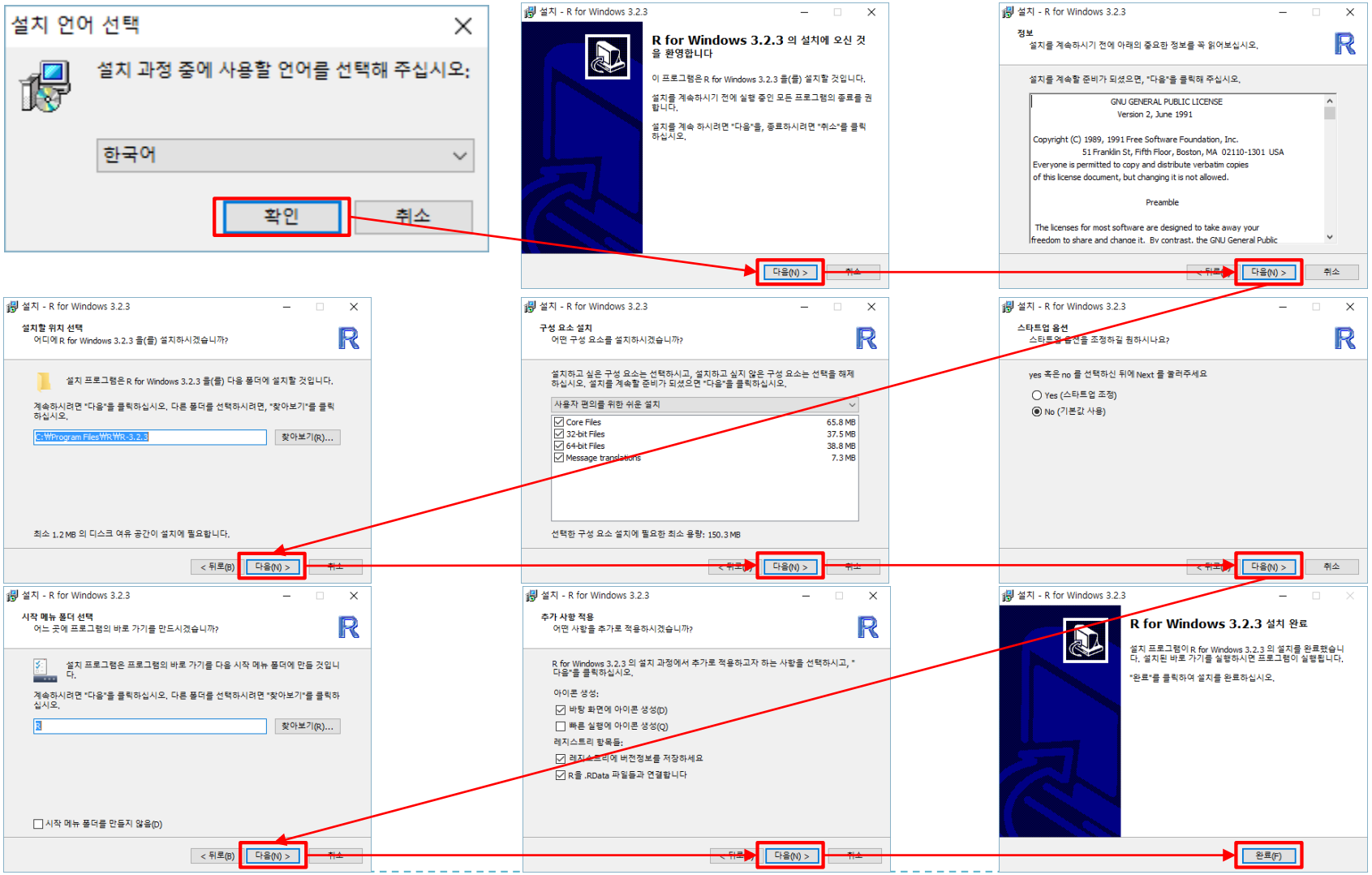
이슈

- 컴퓨터 프로그래밍 언어로 전문가에게는 강력한 도구이지만, 일반 사용자가 사용하기에는 진입장벽이 존재한다.
  - R Community에 다양한 GUI 관련 Project 존재
  - LG CNS SRA 를 통해 개발 편의성 도모
- 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
  - LG CNS SRA 로 관련 기능 지원

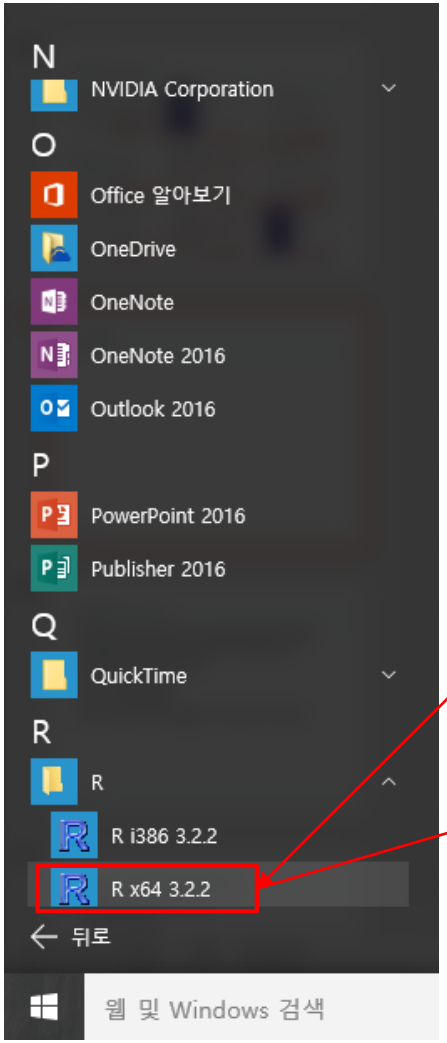
R 설치 (1/2)


https://cran.r-project.org
The Comprehensive R Archive Network
Download and Install R
Precompiled binary distributions of the base system and contributed packages, Windows and Mac users most likely want one of these versions of R:
Download R for Linux
Download R for (Mac) OS X
Download R for Windows
CRAN
Mirrors
What's new?
Task Views
Search
About R
R Homepage
The R Journal
Software
R Sources
R Binaries
Packages
Other
Documentation
Manuals
FAQs
Contributed
R for Windows
Subdirectories:
base
Binaries for base distribution (managed by Duncan Murdoch). This is what you want to install R for the first time.
contrib
Binaries of contributed packages (managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.
Rtools
R-3.2.3 for Windows (32/64 bit)
Please do not submit suggestions related tc
Download R 3.2.3 for Windows
62 megabytes, 32/64 bit)
Installation and other instructions
New features in this version
You may also want to
Note: CRAN does son
If you want to double-check that the package you have downloaded exactly matches the package distributed by R, you can compare the md5sum of the .exe to the true fingerprint. You will need a version of md5sum for windows: both graphical and command line versions are available.
Frequently asked questions
How do I install R when using Windows Vista?
How do I update packages in my previous version of R?
Should I run 32-bit or 64-bit R?
Please see the R FAQ for general information about R and the R Windows FAQ for Windows-specific information.

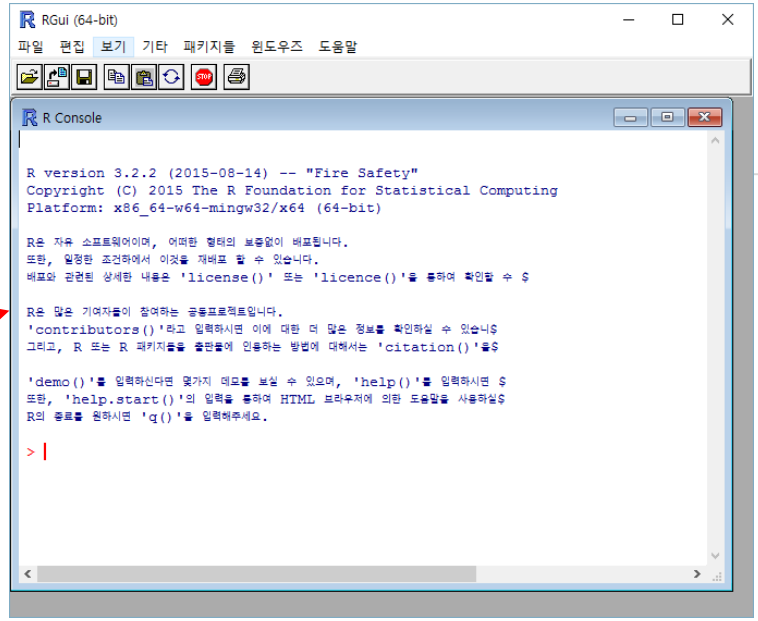
R 설치 (2/2)


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
다음(N) >
취소
정보
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
Preamble
< 뒤로
다음(N) >
취소
설치할 위치 선택
C:\Program Files\R\R-3.2.3
찾아보기(R)...
< 뒤로(B)
다음(N) >
취소
구성 요소 설치
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
스타트업 옵션
yes 혹은 no 를 선택하신 뒤에 Next 를 눌러주세요
Yes (스타트업 조정)
No (기본값 사용)
시작 메뉴 폴더 선택
찾아보기(R)...
시작 메뉴 폴더를 만들지 않음(D)
추가 사항 적용
아이콘 생성:
바탕 화면에 아이콘 생성(D)
빠른 실행에 아이콘 생성(Q)
레지스트리 항목들:
R을 .RData 파일들과 연결합니다
R for Windows 3.2.3 설치 완료
"완료"를 클릭하여 설치를 완료하십시오.
완료(F)

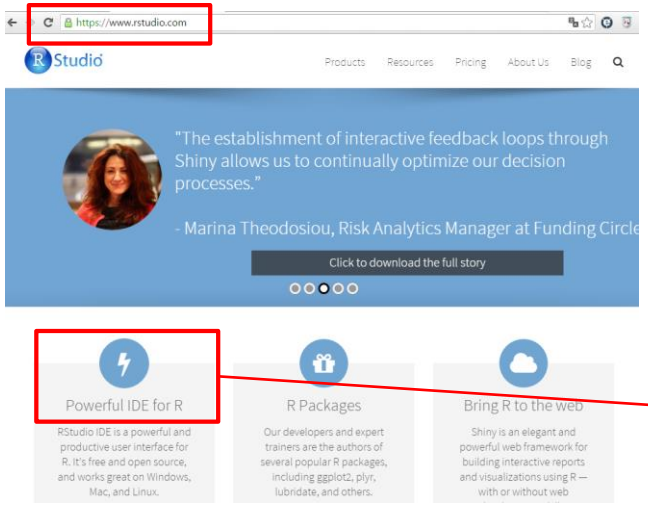
R 실행

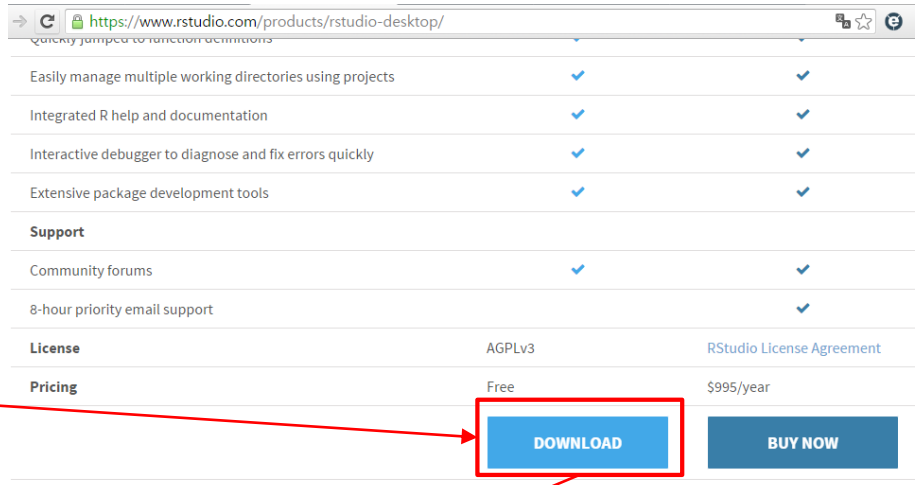

N
NVIDIA Corporation
O
Office 알아보기
OneDrive
OneNote
OneNote 2016
Outlook 2016
P
PowerPoint 2016
Publisher 2016
Q
QuickTime
R
R
R i386 3.2.2
R x64 3.2.2
뒤로
웹 및 Windows 검색

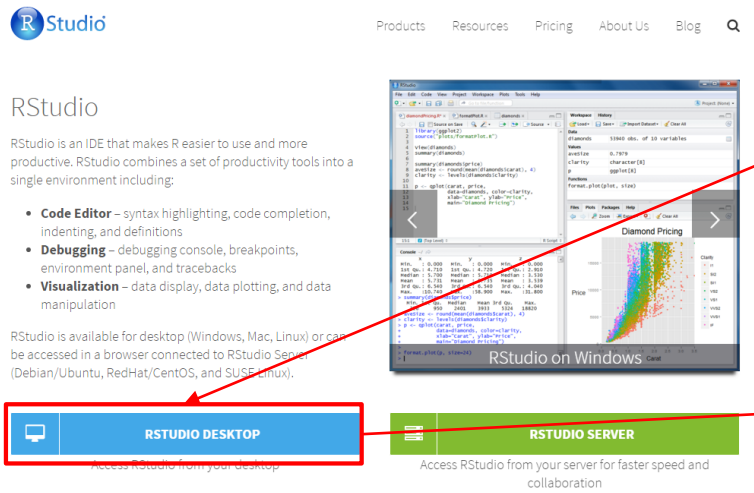

컴퓨터에 대한 기본 정보 보기
Windows 버전
Windows 10 Pro
© 2015 Microsoft Corporation. All rights reserved.
시스템
프로세서: Intel(R) Core(TM) i5 CPU 760 @ 2.80GHz 2.80 GHz
설치된 메모리(RAM): 12.0GB
시스템 종류: 64비트 운영 체제, x64 기반 프로세서
펜 및 터치: 이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다.
컴퓨터 이름, 도메인 및 작업 그룹 설정

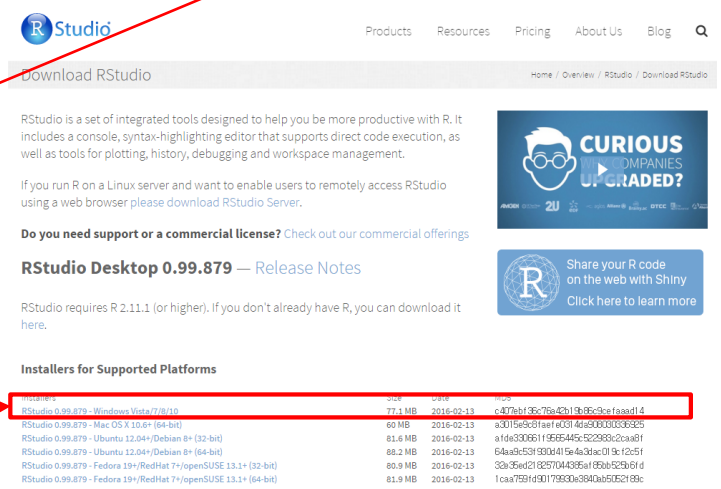

RGui (64-bit)
파일 편집 보기 기타 패키지들 윈도우즈 도움말
R Console
R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R은 자유 소프트웨어이며, 어떠한 형태의 보증없이 배포됩니다.
또한, 일정한 조건하에서 이것을 재배포 할 수 있습니다.
배포와 관련된 상세한 내용은 'license()' 또는 'licence()'을 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)

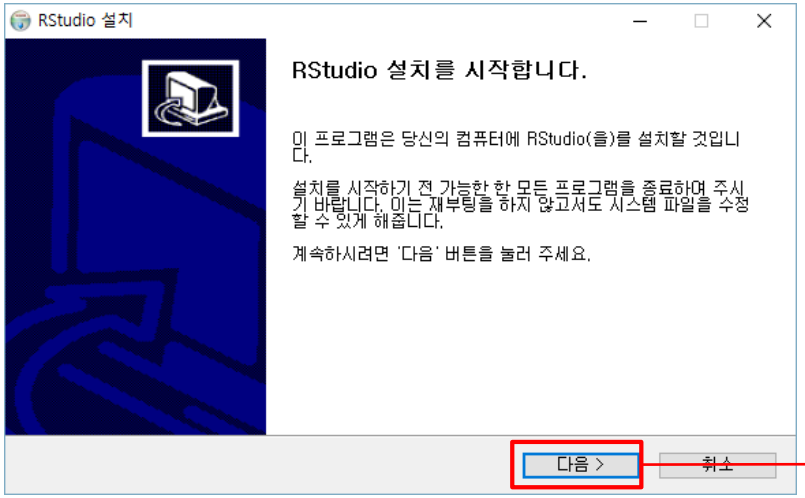


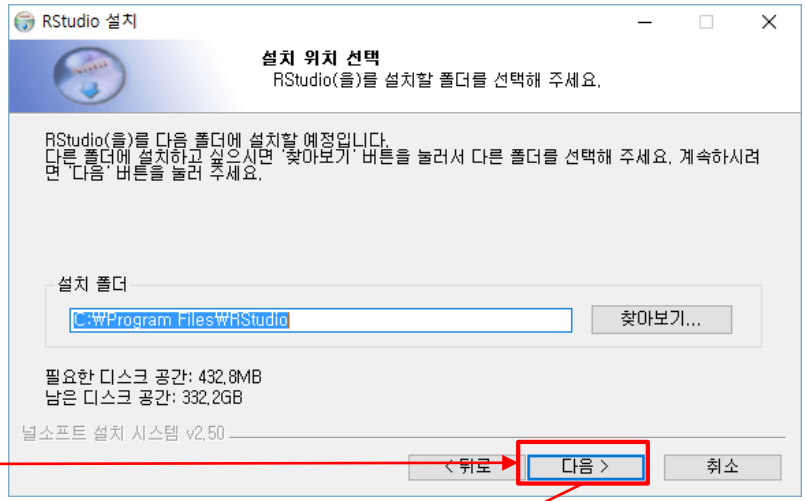


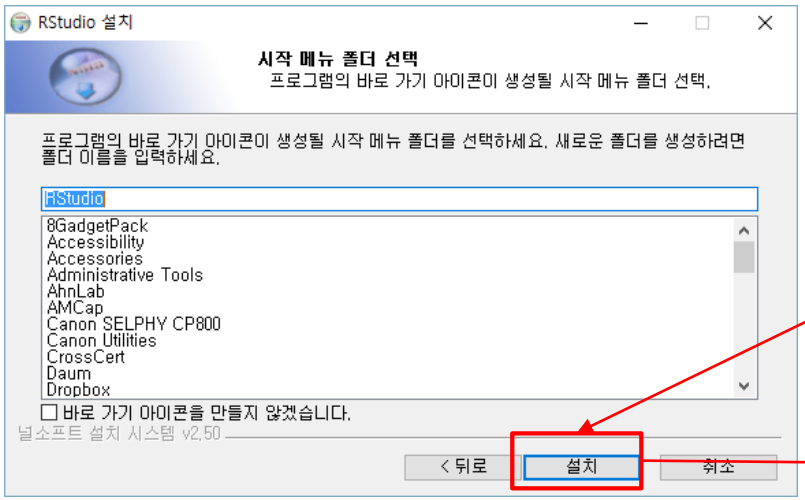


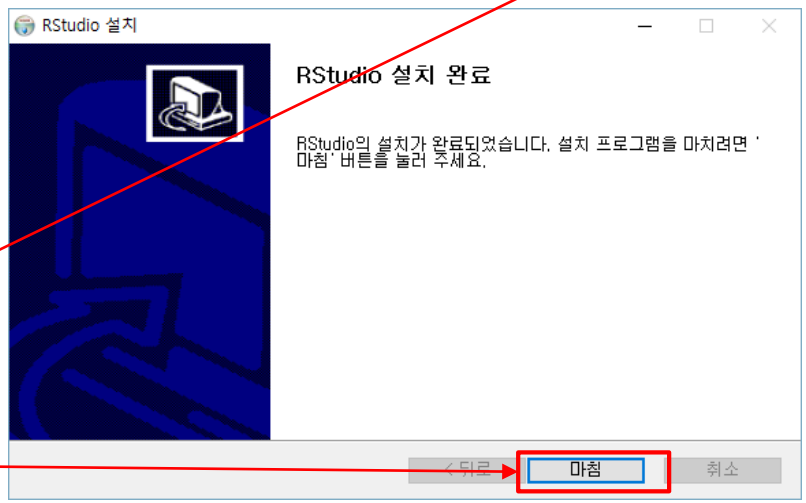


RStudio 설치(2/2)

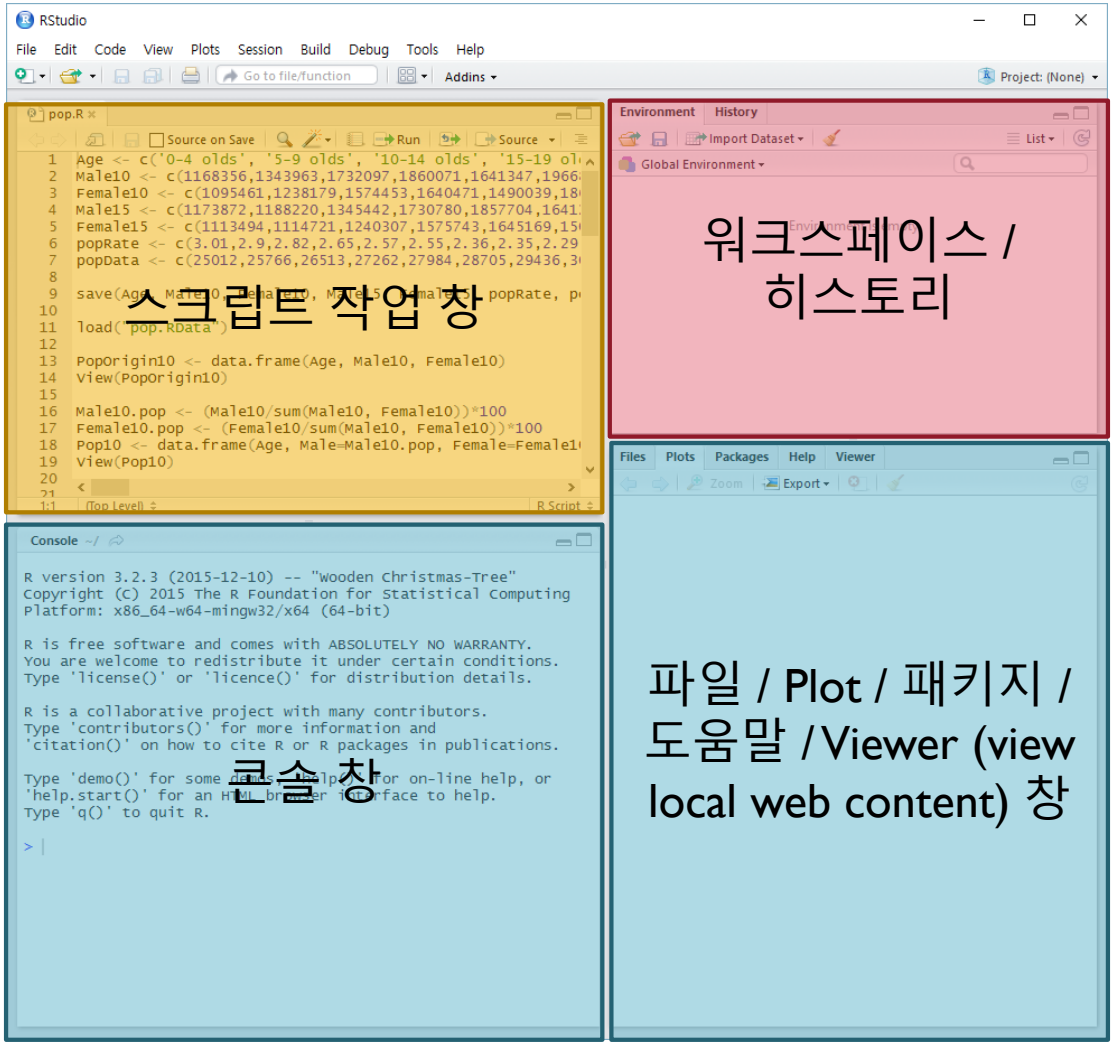








RStudio 화면


RStudio
File Edit Code View Plots Session Build Debug Tools Help
Addins
Project: (None)
pop.R
Source on Save
Run
Source
스크립트 작업 창
Environment History
Import Dataset
List
Global Environment
워크스페이스 /
히스토리
Console
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
콘솔 창
Files Plots Packages Help Viewer
Zoom
Export
파일 / Plot / 패키지 /
도움말 / Viewer (view
local web content) 창

Contents

- 데이터 마이닝 기본
- R 개발 환경
- R의 기본
- 데이터 처리

Object-oriented programming (1/2)

- Class
  - 특정한 attribute을 가지고 있는 objec의 추상적인 정의
  - Dog Class
    - Attributes: color, size, age
- Object
  - 특정 class의 instance
    - Dog1 – color: 'brown', size: 5.5 inch, age: 3 years
- Function
  - 입력 집합을 받아서 출력을 반환하는 절차의 집합
  - 값이 반환되는 것과 값이 반환되지 않는 것 존재

Object-oriented programming (2/2)

- Function
  - Function 구분 방법

```
> var1 <- sum(c(4,3,2))
> var1
[1] 9
```

```
> var2 <- cat(9)
9
> var2
NULL
```

  - Function 정의

```
> test.function <- function(a, b, c){
+ result <- (a * b) + c
+ return(result)
+ }
> test.function (2,3,1)
[1] 7
```

Variables

- 자유로운 Variables 할당
  - Overriding: 동일 variable에 다른 타입의 object 할당 가능

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

  - Variable의 타입 선언 불필요
  - Variable에 값 할당 방법
    - <- or ->: 연결된 값을 variable에 할당
    - =: <-, ->와 동일
    - assign(): 첫번째 parameter가 variable, 두번째 parameter가 값
  - R은 대소문자 구분

Classes (1/6)

- Six atomic classes
  - Character: 문자열, 따옴표로 구분
  - Numeric: 소수 숫자
  - Integer: 정수 숫자
  - Complex: Complex 숫자
  - Logical: TRUE/FALSE
  - Raw: raw bytes

Classes (2/6)

- Vectors
  - 하나의 class만 가지는 요소(element)들의 집합
  - Vector의 type은 element의 type과 동일
  - 기존 type과 다른 element가 추가되면, 에러가 아닌, vector의 type을 변경

```
> aaa <- numeric(length=5)
> aaa[1] <- 6
> aaa[2] <- 2
> class(aaa)
[1] "numeric"
> aaa[3] <- 'a string'
> class(aaa)
[1] "character"
> aaa[1]-aaa[2]
Error in aaa[1] - aaa[2] : non-numeric argument to binary oper
ator
```

Classes (3/6)

- Lists
  - 어떤 class의 어떤 object도 포함하는 vector
  - List는 list를 포함할 수 있음

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

[[2]]
[1] 5

[[3]]
[1] "a string"

> aaa[[2]] - aaa[[1]]
[1] 1
```

Classes (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)
[1] "matrix"
```

  - Matrix는 2 dimension (row, column)을 가지는 array의 한 종류

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

  - Vector와 동일하게 동일 type의 element들만 가질 수 있음

Classes (5/6)

- Data frames
  - List의 특별한 형태: List의 element들의 길이가 동일
  - 2-dimensional matrix와 유사: 다른 class들을 element로 가질 수 있는 것이 다름

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

Classes (6/6)

- Factors
  - Categorical variable을 위한 특별한 class
  - Character element를 나타내는 numeric 코드
  - Integer의 집합과 연관된 level (label)로 이루어짐

```
> animals <- c("dog", "cat", "dog", "horse")
> class(animals)
[1] "character"
> animals
[1] "dog"   "cat"   "dog"   "horse"
>
> animals <- as.factor(animals)
> animals
[1] dog   cat   dog   horse
Levels: cat dog horse
> cat(animals)
2 1 2 3
>
> as.character(animals)
[1] "dog"   "cat"   "dog"   "horse"
> as.numeric(animals)
[1] 2 1 2 3
```

Element selection (1/4)

- Vector
  - By index

```
> LETTERS[c(1,5,6)]
[1] "A" "E" "F"
>
> LETTERS[-c(1,5,6)]
 [1] "B" "C" "D" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P"
[14] "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
>
> rev(LETTERS)
 [1] "Z" "Y" "X" "W" "V" "U" "T" "S" "R" "Q" "P" "O" "N"
[14] "M" "L" "K" "J" "I" "H" "G" "F" "E" "D" "C" "B" "A"
```

  - By name

```
> aaa <- 1:10
> names(aaa) <- letters[1:5]
> names(aaa)
 [1] "a" "b" "c" "d" "e" NA  NA  NA  NA  NA
>
> aaa[c('a','c','e')]
a c e
1 3 5
```

  - By logical

```
> aaa <- 1:5
> aaa[c(T,F,F,T,T)]
[1] 1 4 5
```

  - Recycle

```
> aaa[c(T,F)]
[1] 1 3 5
>
> aaa[c(F,T)]
[1] 2 4
```

  - NA

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

Element selection (2/4)

- Array

```
> aaa <- matrix(1:16, 4, 4)
> aaa
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
>
> aaa[c(1,3), c(2,4)]
     [,1] [,2]
[1,]    5   13
[2,]    7   15
>
> aaa[5]
[1] 5
```

```
> dimnames(aaa) <- LETTERS[1:4]
Error in dimnames(aaa) <- LETTERS[1:4] : 'dimnam
list
>
> dimnames(aaa)[[1]] <- LETTERS[1:4]
> aaa
  [,1] [,2] [,3] [,4]
A    1    5    9   13
B    2    6   10   14
C    3    7   11   15
D    4    8   12   16
```

```
> dimnames(aaa)[[2]] <- letters[1:4]
> aaa
  a b  c  d
A 1 5  9 13
B 2 6 10 14
C 3 7 11 15
D 4 8 12 16
> row.names(aaa) <- LETTERS[1:4]
> colnames(aaa) <- letters[1:4]
```

```
> aaa[c('A','C'), c('a','d')]
  a  d
A 1 13
C 3 15
>
> aaa[1:2,]
  a b  c  d
A 1 5  9 13
B 2 6 10 14
```

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])
[1] "list"
>
> class(list.ex[[2]])
[1] "character"
>
> list.ex[['b']]
[1] "a" "b" "c"
>
> list.ex[[1:3]]
Error in list.ex[[1:3]] : recursive indexing failed at level 2
>
> list.ex$a
[1] 1 2 3
```

Element selection (4/4)

- Data frame

```
> test.data.frame <- data.frame(Var1=1:10, Var2
=LETTERS[1:10])
> test.data.frame$Var1
 [1]  1  2  3  4  5  6  7  8  9 10
> test.data.frame[['Var1']]
 [1]  1  2  3  4  5  6  7  8  9 10
> test.data.frame[[1]]
 [1]  1  2  3  4  5  6  7  8  9 10
```

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

```
> subset(test.data.frame, Var1 >= 8)
   Var1 Var2
8     8    H
9     9    I
10   10    J
```

```
> test.data.frame <- data.frame(Var1 = 1:10, Va
r2 = LETTERS[1:10], Var3 = LETTERS[11:20])
> subset(test.data.frame, Var1 >=8, select = c(
Var1, Var3))
   Var1 Var3
8     8    R
9     9    S
10   10    T
>
> subset(test.data.frame, Var1 >=8, select = -V
ar2)
   Var1 Var3
8     8    R
9     9    S
10   10    T
```

```
> idx <- which(test.data.frame$Var1 >= 8)
> test.data.frame[idx,]
   Var1 Var2 Var3
8     8    H    R
9     9    I    S
10   10    J    T
```

Control Structure (1/2)

- if … else

```
> a <- 5
> if (a > 0) {print ('a is greater than 0')} else
+ {print ('a is smaller than 0')}
[1] "a is greater 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 0')}
[1] "a is greater than 0"
```

```
> if (a <0) { print ('a')}
> else {print ('b')}
Error: unexpected 'else' in "else"
>
> if (a < 0) {print ('a')
+ } else {print ('b')}
[1] "b"
```

- while

```
> a <- 1
> while (a < 4) {
+ print (paste ('This is iteration', a))
+ a <- a + 1
+ }
[1] "This is iteration 1"
[1] "This is iteration 2"
[1] "This is iteration 3"
```

Control Structure (2/2)

- for

```
> vector <- c('aaa', 'bbb', 'ccc')
> for (i in vector) {
+ print(i)
+ }
[1] "aaa"
[1] "bbb"
[1] "ccc"
>
> num <- c(1:3)
> for (i in num) {
+ print (i+1)
+ }
[1] 2
[1] 3
[1] 4
```

- switch

```
> inp <- 'b'
> switch(inp,
+ a = print ('inp is a'),
+ b = print ('inp is b'),
+ c = print ('inp is c'))
[1] "inp is b"
>
> inp <- 'd'
> switch(inp,
+ a = print ('inp is a'),
+ b = print ('inp is b'),
+ c = print ('inp is c'),
+ print('inp is not a, b, c'))
[1] "inp is not a, b, c"
```

```
> inp <- 1
> switch(inp,
+ 1 = print(inp + 1),
Error: unexpected '=' in:
"switch(inp,
1 ="
```

```
> inp <- 1
> switch(inp,
+ '1' = print(inp + 1),
+ print('none'))
[1] 2
```

```
> inp <- 2
> switch(inp,'inp+1', 'inp+2', 'inp+3', 'inp+4')
[1] "inp+2"
```

Reading data (1/3)

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

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

- Line by line

```
> path <- 'https://archive.ics.uci.edu/ml/machine
-learning-databases/iris/iris.data'
> data <- readLines(path)
>
> class(data)
[1] "character"
> length(data)
[1] 151
```

- Character set

```
> path <- 'https://archive.ics.uci.edu/ml/machine
-learning-databases/iris/iris.data'
> data <- readChar(path, nchars = 1e5)
Warning message:
In readChar(path, nchars = 1e+05) :
  can only read in bytes in a non-UTF-8 MBCS loca
le
>
> class(data)
[1] "character"
> length(data)
[1] 1
```

Reading data (2/3)

- JSON
  - JavaScript Object Notation
  - RJSONIO

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

  - rjson

```
> url <- 'http://api.worldbank.org/v2/datacatalog
?format=json'
>
> raw.json <- readChar(url, nchars=1e6)
Warning message:
In readChar(url, nchars = 1e+06) :
  can only read in bytes in a non-UTF-8 MBCS loca
le
>
> json <- fromJSON(raw.json)
```

```
> library(rjson)
Error in nchar(homeDir) : invalid multibyte strin
g, element 1

다음의 패키지를 부착합니다: 'rjson'

The following objects are masked from 'package:RJ
SONIO':

    fromJSON, toJSON
```

- XML

```
> library(XML)
>
> url <- 'http://api.worldbank.org/v2/datacatalog
?format=xml'
>
> xml.obj <- xmlTreeParse(url)
>
> class(xml.obj)
[1] "XMLDocument"         "XMLAbstractDocument"
```

Reading data (3/3)

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

Contents

- 데이터 마이닝 기본
- R 개발 환경
- R의 기본
- 데이터 처리

View

- Size and Structure

```
> dim(iris)
[1] 150   5
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length"
[4] "Petal.Width"  "Species"
> str(iris)
'data.frame':   150 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/ 3 levels "setosa","versic
olor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

- Attributes of Data

```
> attributes(iris)
$names
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length"
[4] "Petal.Width"  "Species"

$row.names
  [1]   1   2   3   4   5   6   7   8   9  10  11
 [12]  12  13  14  15  16  17  18  19  20  21  22
 [23]  23  24  25  26  27  28  29  30  31  32  33
 [34]  34  35  36  37  38  39  40  41  42  43  44
 [45]  45  46  47  48  49  50  51  52  53  54  55
 [56]  56  57  58  59  60  61  62  63  64  65  66
 [67]  67  68  69  70  71  72  73  74  75  76  77
 [78]  78  79  80  81  82  83  84  85  86  87  88
 [89]  89  90  91  92  93  94  95  96  97  98  99
[100] 100 101 102 103 104 105 106 107 108 109 110
[111] 111 112 113 114 115 116 117 118 119 120 121
[122] 122 123 124 125 126 127 128 129 130 131 132
[133] 133 134 135 136 137 138 139 140 141 142 143
[144] 144 145 146 147 148 149 150

$class
[1] "data.frame"
```

Explore

- Summary

```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length
 Min.   :4.300   Min.   :2.000   Min.   :1.000
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600
 Median :5.800   Median :3.000   Median :4.350
 Mean   :5.843   Mean   :3.057   Mean   :3.758
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100
 Max.   :7.900   Max.   :4.400   Max.   :6.900
  Petal.Width          Species
 Min.   :0.100   setosa     :50
 1st Qu.:0.300   versicolor:50
 Median :1.300   virginica :50
 Mean   :1.199
 3rd Qu.:1.800
 Max.   :2.500
```

- Mean, Median, Range, Quantile

```
> range(iris$Sepal.Length)
[1] 4.3 7.9
> quantile(iris$Sepal.Length)
  0%  25%  50%  75% 100%
 4.3  5.1  5.8  6.4  7.9
> quantile(iris$Sepal.Length, c(0.1, 0.3, 0.65))
 10%  30%  65%
4.80 5.27 6.20
```

- Variance, Histogram

```
> var(iris$Sepal.Length)
[1] 0.6856935
> hist(iris$Sepal.Length)
```

sorting

- sort()

```
> vec1 <- c(3, 2, 5, 1, 4)
> sort(vec1)
[1] 1 2 3 4 5
>
> sort(c(T, T, F, F))
[1] FALSE FALSE  TRUE  TRUE
>
> sort(c('play', 'plan', 'plot', 'proof'))
[1] "plan"  "play"  "plot"  "proof"
```

- order()

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

- sort() vs order()

```
> vec1[order(vec1)]
[1] 1 2 3 4 5

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


> data("iris")
> names(iris)
[1] "Sepal.Length" "Sepal.Width"
[3] "Petal.Length" "Petal.Width"
[5] "Species"
> iris.ordered <- iris[order(iris$Sepal.Length, i
ris$Sepal.Width),]
```

summary

- table()

```
> sample <- data.frame(var1=rep(c('Male', 'Female
'), 10), var2 = rep(c('A', 'B', 'C', 'D')))
> example.table <- table(sample$var1, sample$var2
)
> example.table

        A B C D
  Female 0 5 0 5
  Male   5 0 5 0
```

- aggregate()
  - vector

```
> data(iris)
> aggregate(iris$Sepal.Length, by=list(iris$Speci
es), FUN='mean')
     Group.1     x
1     setosa 5.006
2 versicolor 5.936
3  virginica 6.588
```

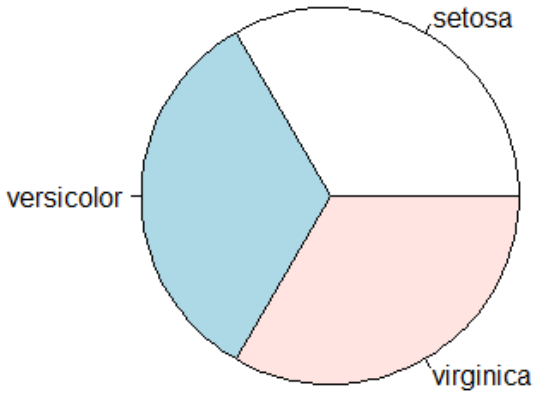
  - object

```
> aggregate(iris$Sepal.Length ~ iris$Species, FUN
='mean')
  iris$Species iris$Sepal.Length
1       setosa              5.006
2   versicolor              5.936
3    virginica              6.588
> aggregate(Sepal.Length ~ Species, data=iris, FU
N='mean')
     Species Sepal.Length
1     setosa        5.006
2 versicolor        5.936
3  virginica        6.588

> iris$letter <- letters[1:2]
> aggregate(Sepal.Length ~ Species + letter, data
=iris, FUN='mean')
     Species letter Sepal.Length
1     setosa      a        5.024
2 versicolor      a        5.992
3  virginica      a        6.504
4     setosa      b        4.988
5 versicolor      b        5.880
6  virginica      b        6.672

> aggregate(cbind(Sepal.Length, Sepal.Width) ~ Sp
ecies, data = iris, FUN = 'mean')
     Species Sepal.Length Sepal.Width
1     setosa        5.006       3.428
2 versicolor        5.936       2.770
3  virginica        6.588       2.974

> data(iris)
> aggregate(. ~ Species, data=iris, FUN='mean')
```

Summary Chart

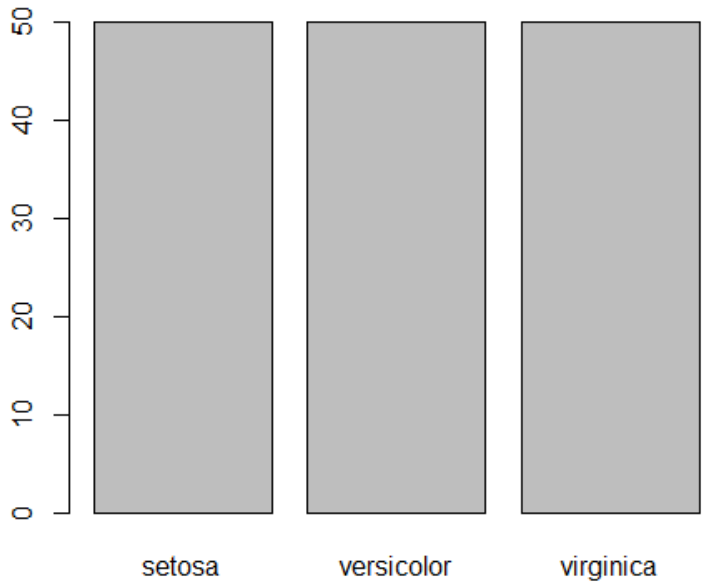
- Pie Chart

```
> table(iris$Species)

    setosa versicolor  virginica
        50         50         50
> pie(table(iris$Species))
```



- Bar Chart

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



grep

- grep()
  - vecor에서 elemen의 인덱스를 반환
- grepl()
  - 입력 vector와 동일한 길이의 logical vector 반환
- gsub()
  - 패턴을 찾아서 설정된 값으로 변경
- gregexpr()
  - 패턴과 일치하는 지점의 list 반환

Regular expression (1/4)

- Set

```
> gregexpr('[a-z]', 'string 01 A')
[[1]]
[1] 1 2 3 4 5 6
attr(,"match.length")
[1] 1 1 1 1 1 1
attr(,"useBytes")
[1] TRUE

> gregexpr('[a-z0]', 'string 01 A')
[[1]]
[1] 1 2 3 4 5 6 8
attr(,"match.length")
[1] 1 1 1 1 1 1 1
attr(,"useBytes")
[1] TRUE
```

- Shortcut
  - http://stat.ethz.ch/R-manual/R-devel/library/base/html/regex.html

```
> gregexpr('[[:alnum:]]', 'string 01 A')
[[1]]
[1]  1  2  3  4  5  6  8  9 11
attr(,"match.length")
[1] 1 1 1 1 1 1 1 1 1
attr(,"useBytes")
[1] TRUE
```

- Dot

```
> gregexpr('.', 'string 01 A')
[[1]]
 [1]  1  2  3  4  5  6  7  8  9 10 11
attr(,"match.length")
 [1] 1 1 1 1 1 1 1 1 1 1 1
attr(,"useBytes")
[1] TRUE
```

Regular expression (2/4)

- Non-printable char

```
> gregexpr('\\n', 'string 01
+             A')
[[1]]
[1] 11
attr(,"match.length")
[1] 1
attr(,"useBytes")
[1] TRUE
```

- Negation

```
> gregexpr('[^a-z]', 'string 01 A')
[[1]]
[1]  7  8  9 10 11
attr(,"match.length")
[1] 1 1 1 1 1
attr(,"useBytes")
[1] TRUE
```

- Alternation

```
> gregexpr('r|n', 'string 01 A')
[[1]]
[1] 3 5
attr(,"match.length")
[1] 1 1
attr(,"useBytes")
[1] TRUE
```

- Quantifier
  - {min, max}
  - *, ?, +

```
> gregexpr('[a-z]{2,4}', 'string 01 A')
[[1]]
[1] 1 5
attr(,"match.length")
[1] 4 2
attr(,"useBytes")
[1] TRUE
```

Regular expression (3/4)

- Anchor

```
> grep('tri', c('string', 'triangle'), value=T)
[1] "string"   "triangle"
> grep('^tri', c('string', 'triangle'), value=T)
[1] "triangle"
>
> grep('^triangle', c('string', 'triangle', 'tria
ngles'), value=T)
[1] "triangle"  "triangles"
> grep('^triangle$', c('string', 'triangle', 'tri
angles'), value=T)
[1] "triangle"
```

- Escape
  - ., ?, + 등

```
> gregexpr('\\?', 'what is this?')
[[1]]
[1] 13
attr(,"match.length")
[1] 1
attr(,"useBytes")
[1] TRUE
```

- Expression

```
> grep('^pri|tri', c('triangle', 'triangles', 'pr
ice', 'priority', 'string'), value=T)
[1] "triangle"  "triangles" "price"
[4] "priority"  "string"
> grep('^(pri|tri)', c('triangle', 'triangles', '
price', 'priority', 'string'), value=T)
[1] "triangle"  "triangles" "price"
[4] "priority"
```

Regular expression (4/4)

```
> numbers <- c('+1 456-2341', '5342673', '55578274982', '74683029873', '25', '+442 5421611')
>
> grep('^(\\+[0-9]+\\s)?[0-9]{3}\\-?[0-9]{4}$', numbers, value=T)
[1] "+1 456-2341"  "5342673"      "+442 5421611"
>
> grep('^(\\+[0-9]+\\s)?[0-9]{3}\\-?[0-9]{4}', numbers, value=T)
[1] "+1 456-2341"  "5342673"      "55578274982"  "74683029873"  "+442 5421611"
>
>
> example <- 'This is a text. What is this exactly? A text. Are you sure?'
>
> greps <- gregexpr('\\s[A-Z]{1}[^\\?\\.]*\\?', example)
>
> regmatches(example, greps)
[[1]]
[1] " What is this exactly?" " Are you sure?"

> greps <- gregexpr('\\s[A-Z]{1}.*\\?', example)
>
> regmatches(example, greps)
[[1]]
[1] " What is this exactly? A text. Are you sure?"
```

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)
      a.75%       b.75%       c.75%
  0.7637704  77.0660968 146.5353989
>
> sapply(sample.list, function(x) round(sum(x+2)))
    a     b     c
  252 26846  3845
>
> sapply(sample.list, function(x) quantile(x, probs=0.75))
      a.75%       b.75%       c.75%
  0.7637704  77.0660968 146.5353989


> vapply(sample.list, quantile, probs=0.75, c('Mean' = 0))
          a           b           c
  0.7637704  77.0660968 146.5353989
> lapply(sample.list, quantile, probs=0.75)
$a
      75%
0.7637704

$b
    75%
77.0661

$c
     75%
146.5354
```

```
> x <- as.numeric(c(1:1000000))
> ptm <- proc.time()
> for (i in x) X[i] <- x[i] * x[i]
> proc.time() - ptm
 사용자  시스템 elapsed
   2.87    0.00    2.91
> ptm <- proc.time()
> X <- sapply(x, function(x) x * x)
> proc.time() - ptm
 사용자  시스템 elapsed
   1.59    0.03    1.62
```

plyr

- Tools for splitting, applying, and combining data.

```
> library(plyr)
> ddply(iris, .(Species), summarize, indicator1=quantile(Sepal.Length, 0.75),
indicator2=sum(Sepal.Width)/sum(Petal.Length))
     Species indicator1 indicator2
1     setosa        5.2  2.3447332
2 versicolor        6.3  0.6502347
3  virginica        6.9  0.5356628
>
> dlply(iris, .(Species), summarize, indicator1=quantile(Sepal.Length, 0.75),
indicator2=sum(Sepal.Width)/sum(Petal.Length))
$setosa
  indicator1 indicator2
1        5.2   2.344733

$versicolor
  indicator1 indicator2
1        6.3  0.6502347

$virginica
  indicator1 indicator2
1        6.9  0.5356628

attr(,"split_type")
[1] "data.frame"
attr(,"split_labels")
     Species
1     setosa
2 versicolor
3  virginica
```

reshape2

- melt(), dcast(), acast()

```
> library(reshape2)
> data(iris)
> melt(iris)
Using Species as id variables
        Species     variable value
1        setosa Sepal.Length   5.1
2        setosa Sepal.Length   4.9
3        setosa Sepal.Length   4.7
4        setosa Sepal.Length   4.6
5        setosa Sepal.Length   5.0
6        setosa Sepal.Length   5.4
7        setosa Sepal.Length   4.6
8        setosa Sepal.Length   5.0
9        setosa Sepal.Length   4.4
10       setosa Sepal.Length   4.9
11       setosa Sepal.Length   5.4
12       setosa Sepal.Length   4.8
13       setosa Sepal.Length   4.8
14       setosa Sepal.Length   4.3
15       setosa Sepal.Length   5.8
16       setosa Sepal.Length   5.7
17       setosa Sepal.Length   5.4
18       setosa Sepal.Length   5.1
19       setosa Sepal.Length   5.7
20       setosa Sepal.Length   5.1
```

```
> data("iris")
> m.iris <- melt(iris)
Using Species as id variables
> dcast(m.iris, variable~Species, fun.aggregate=sum)
      variable setosa versicolor virginica
1 Sepal.Length  250.3      296.8     329.4
2  Sepal.Width  171.4      138.5     148.7
3 Petal.Length   73.1      213.0     277.6
4  Petal.Width   12.3       66.3     101.3
>
> dcast(m.iris, variable~Species, fun.aggregate=func
tion(x) sum(x+2))
      variable setosa versicolor virginica
1 Sepal.Length  350.3      396.8     429.4
2  Sepal.Width  271.4      238.5     248.7
3 Petal.Length  173.1      313.0     377.6
4  Petal.Width  112.3      166.3     201.3
```

```
> class(acast(m.iris, variable~Species, fun.aggregat
e=sum))
[1] "matrix"
> class(dcast(m.iris, variable~Species, fun.aggregat
e=sum))
[1] "data.frame"
```

Save to Files

- Charts

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

- Text

```
> write.csv(iris, file='iris.txt')
```

- RData

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