

빅데이터의 이해와 활용

Understanding and Using Big Data

05

R 프로그램 기초

학습 내용

- 01 R 프로그램 개요
- 02 데이터의 입력과 출력
- 03 데이터 구조

학습 목표

- R 프로그램의 특징을 이해하고
R 프로그램의 설치 방법을 설명할 수 있다.
- R 프로그램을 이용하여 데이터를
읽고 쓰는 방법을 설명할 수 있다.
- R 프로그램에서 사용하는 데이터 구조를
세분화하여 설명할 수 있다.

생각 해보기

오픈소스 (Opensource)





01

R 프로그램 개요

- 1) R 프로그램의 개요
- 2) R 프로그램 설치
- 3) R의 실행
- 4) 변수의 활용

1) R 프로그램의 개요

- R 프로그램이란?

R 프로그램

데이터 분석과 시각화 등을 위해 개발된 오픈소스 프로그램



로스 이하카



로버트 젠틀맨

- 벨 연구소에서 개발된 S언어를 바탕으로 만들어짐

- 뉴질랜드 오클랜드 대학의 로스 이하카와 로버트 젠틀맨에 의해 개발됨

- 현재는 R 코어팀에 의해 지속적으로 유지되고 있음

1) R 프로그램의 개요

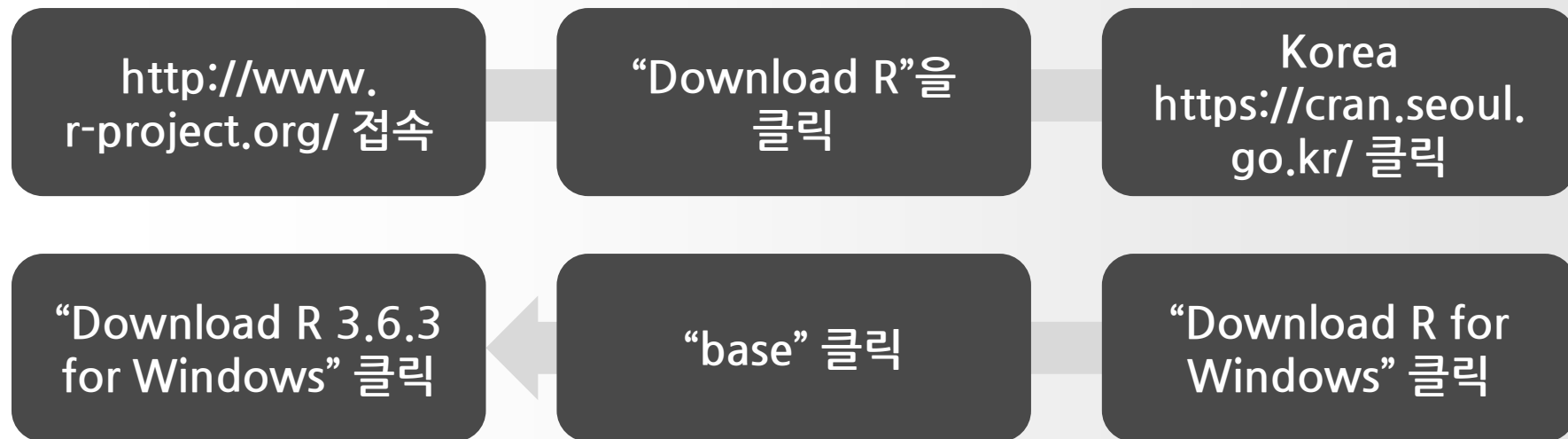
● R 프로그램의 특징

- 01 통계, 머신 러닝, 금융, 시각화 등의 다양한 패키지를 갖추고 있으며 무료로 제공됨
- 02 오픈 소스로 수많은 R 사용자들이 자유롭게 분석 기법을 추가하는 것이 가능함
- 03 Windows, MacOS, Linux 등 여러 운영체제에서 구동이 가능함
- 04 메모리 기반으로 동작됨
- 05 도움말 기능이 뛰어나므로 새로운 기능을 익히기가 쉬움

2) R 프로그램 설치

- R의 설치

“ R은 CRAN(the Comprehensive R Archive Network)을
통해 최신 버전을 다운로드 할 수 있음 ”



※ 사용자 계정은 반드시 영문으로 되어 있을 것

2) R 프로그램 설치

- R Studio의 설치

R Studio

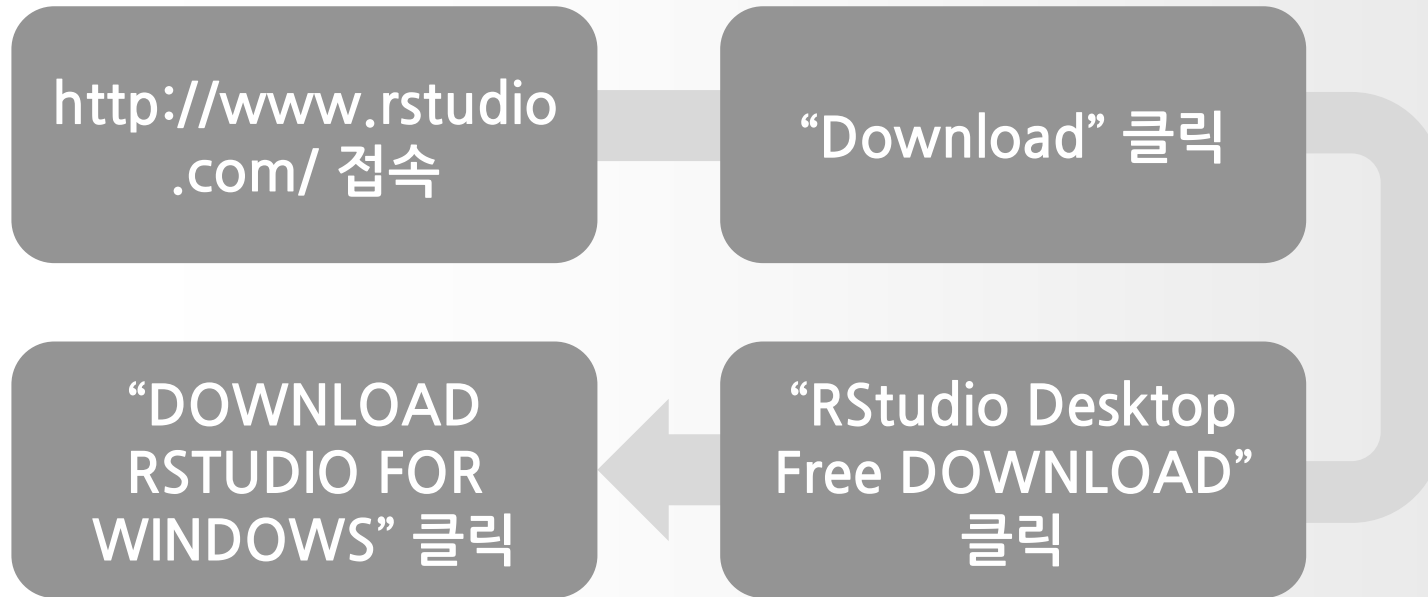
R을 편리하게 사용할 수 있는 사용자 인터페이스를 제공하는 프로그램

↳ 명령 콘솔 이외에도 파일 편집, 데이터 보기, 명령 히스토리 등을 제공함



2) R 프로그램 설치

- R Studio의 설치



2) R 프로그램 설치

● R 프로그램의 화면



R 프로그램은 바탕화면에 생성된 아이콘을 더블 클릭하여 실행함



R 콘솔(Console) 창은 대화식 화면으로 프롬프트 뒤에 명령어를 입력하고 [Enter]를 눌러 실행함

```
RGui (64-bit)
파일 편집 보기 기타 패키지를 윈도우즈 도움말

R Console

R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R은 자유 소프트웨어이며, 이러한 형태의 배포권이 배포됩니다.
또한, 일정한 조건하에서 이것을 재배포 할 수 있습니다.
배포에 관련된 상세한 내용은 'license()' 또는 'licence()'를 통하여 확인할 수 있습니다.

R은 많은 기여자들이 참여하는 공동프로젝트입니다.
'contributors()'라고 입력하시면 이에 대한 더 많은 정보를 확인할 수 있습니다.
그리고, R 또는 R 패키지를 출판물에 인용하는 방법에 대해서는 'citation()'을 참조하십시오.

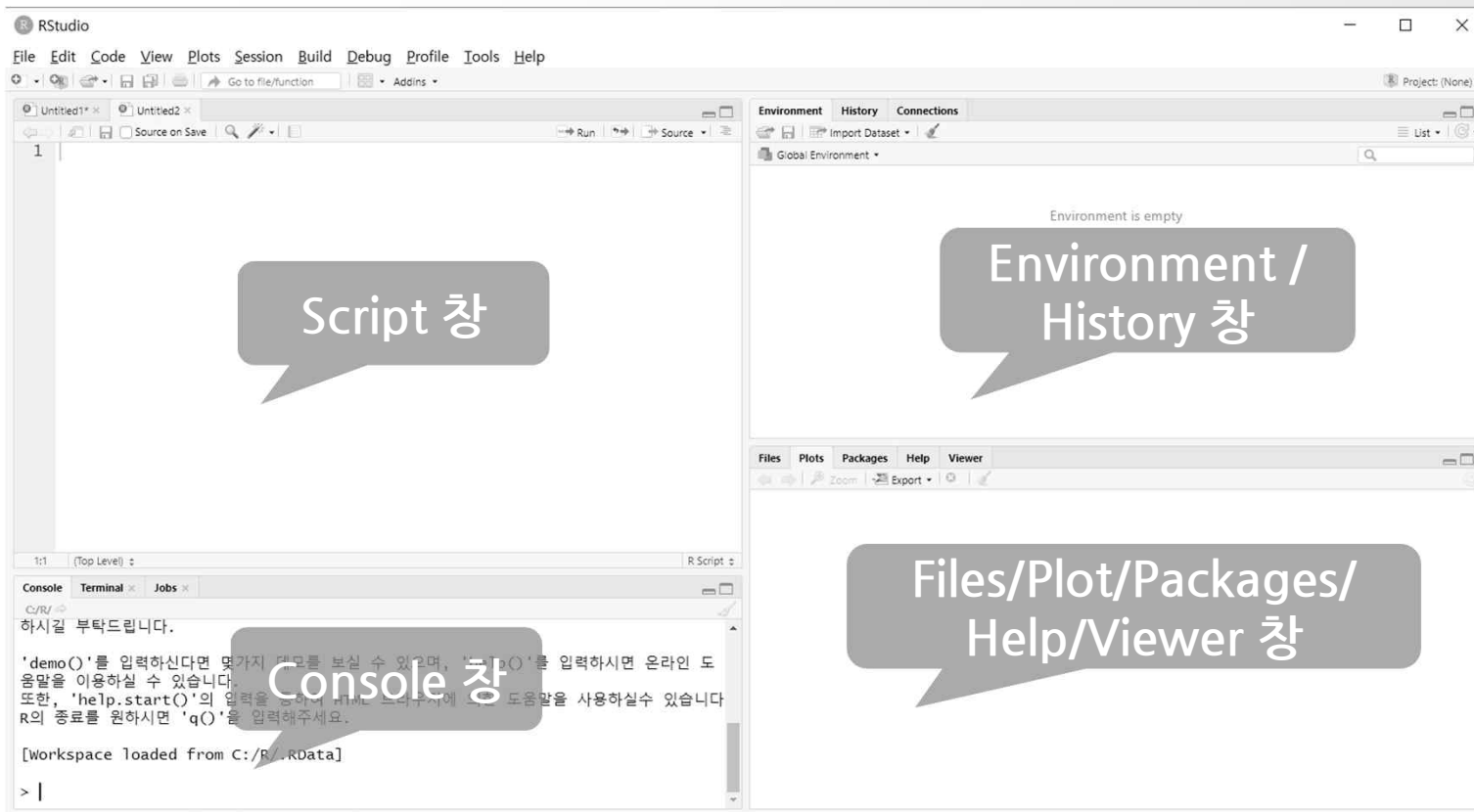
'demo()'를 입력하신다면 몇가지 예제를 보실 수 있으며, 'help()'를 입력하시면 R
또한, 'help.start()'의 링크를 통하여 HTML 브라우저의 R 도움말을 사용할 수 있습니다.
R의 종료는 'q()'를 입력하십시오.

> |
```

2) R 프로그램 설치

- R Studio의 화면

“ 다음과 같이 구성되어 있음 ”



환경 설정

Tools-Global
Options

3) R의 실행

- 명령문 실행

한 줄만 실행 시

Ctrl + Enter

여러 줄 실행 시

블록 지정 후 Ctrl + Enter

여러 명령어를
한 줄에 입력 시

세미콜론(;)으로 구분함

주석문

#을 이용함

3) R의 실행

- 명령문 실행

작업 디렉토리 확인

`getwd()`

작업 디렉토리 지정

`setwd("D:/Rdata")`

도움말

? 명령어 또는 `help(명령어)`

3) R의 실행

- 명령문 실행

13+2

▶ # 더하기

13-2

▶ # 빼기

13*2

▶ # 곱하기

13/2

▶ # 나누기

13%%2

▶ # 나머지

3) R의 실행

- 명령문 실행

```
sum(1:3);mean(1:3)
```

▶ # 여러 명령어

```
getwd()
```

▶ # 작업 디렉토리 확인

```
setwd("D:/Rdata")
```

▶ # 작업 디렉토리 지정

```
? getwd
```

▶ # 도움말

```
help(sum)
```

▶ # 도움말

4) 변수의 활용

● 변수의 활용



문자 a~z, A~Z, 숫자 0~9, _(언더스코어), .(마침표)의 조합으로 구성함



첫 글자는 알파벳 또는 .(마침표)로 시작함



대문자와 소문자는 서로 구분함



변수에 값을 할당하려면 '<->' 또는 '=' 기호를 이용함

4) 변수의 활용

- 변수의 활용

```
x <- 5
```

```
a <- "hello"
```

```
y <- -3
```

```
a
```

```
z <- sqrt(9)
```

```
b <- 'good'
```

```
x+y-z
```

```
b
```

4) 변수의 활용

- R 패키지

R 패키지

특정 분석을 수행할 수 있는 함수, 객체, 데이터, 도움말 등의
집합

↳ R에서는 CRAN이라는 서버에 패키지를 저장하고
사용자에게 제공함



4) 변수의 활용

● R 패키지

패키지 설치

`install.packages`
("패키지명") 명령문을
사용함

패키지 활성화

`library(패키지명)`
명령문을 사용함



4) 변수의 활용

● R 패키지

```
library()
```

▶ # 설치된 라이브러리 확인

```
install.packages  
("ggplot2")
```

▶ # 패키지 설치

```
library(ggplot2)
```

▶ # 패키지 활성화

The background features a dark, abstract design with glowing white lines forming a network of triangles and polygons. A prominent white hexagon is located in the upper center. On the left side, there are several overlapping, white-outlined diamond shapes. The overall aesthetic is futuristic and technical.

02

데이터의 입력과 출력

- 1) 기본 데이터 사용
- 2) 데이터의 입력
- 3) 데이터의 출력

1) 기본 데이터 사용

- R이 제공하는 기본 데이터

“ R에서는 여러 분야에서
수집한 데이터를 기본으로 제공함 ”

data()

▶ # R이 제공하는 기본 데이터 목록

iris

▶ # 붓꽃의 품종에 관한 데이터

str(iris)

▶ 데이터의 형식과 내용을 요약

1) 기본 데이터 사용

- R이 제공하는 기본 데이터

“

R에서는 여러 분야에서
수집한 데이터를 기본으로 제공함

”

women



미국 30대 여성의
평균 키와 몸무게 데이터

str(women)



데이터 구조, 변수 개수, 변수명,
관찰치 개수, 관찰치의 미리보기

2) 데이터의 입력

- 키보드를 이용한 데이터의 입력

“ c() 함수를 이용하여 데이터를 입력함 ”

```
v1 <- c(10,30,80,20)
```

▶ # 벡터 v1에
10, 30, 80, 20을 입력함

```
v1
```

▶ # v1을 출력

2) 데이터의 입력

- 외부 파일을 이용한 데이터 입력



read.table() 함수를 이용하여 데이터 불러오기

- 변수명이 있는 경우 header=T라는 옵션을 지정함

```
data1 <- read.table("ex-1.txt",header = T)  
data1
```

2) 데이터의 입력

● 외부 파일을 이용한 데이터 입력



read.csv() 함수를 이용하여 데이터 불러오기

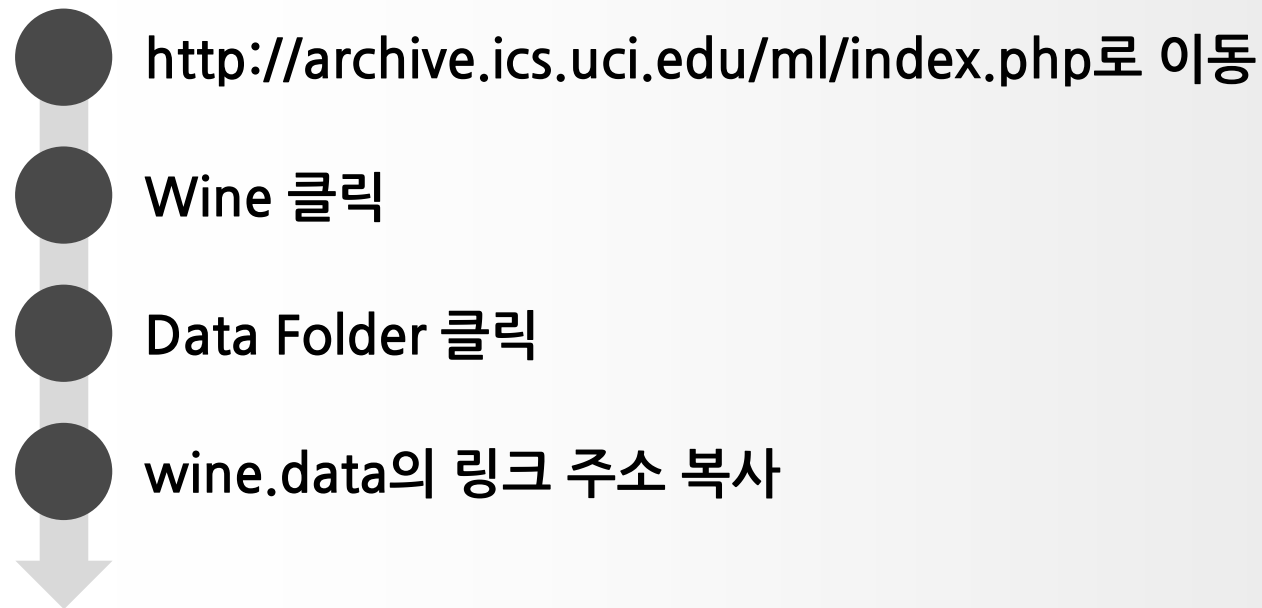
- CSV는 구분기호로 콤마(,)를 사용하는 파일의 형식으로 Excel에서 저장할 수 있는 형식으로 많이 사용됨
- read.csv() 함수는 header=T 옵션을 디폴트로 가짐

```
data2 <- read.csv("ex-2.csv")  
data2
```

2) 데이터의 입력

- 외부 파일을 이용한 데이터 입력

웹사이트에서 데이터 불러오기



2) 데이터의 입력

- 외부 파일을 이용한 데이터 입력

```
website="http://archive.ics.  
uci.edu/ml/machine-learning-  
databases/wine/wine.data"
```

```
data3 <- read.table  
(website,header=T,sep=",")
```

```
data3
```

```
View(data3)
```

▶ # 해당 데이터셋 출력

3) 데이터의 출력

● 데이터의 출력



write.table() 함수를 이용한 데이터 출력

- 생성된 데이터를 외부 파일로 저장함

```
write.table(iris,"iris1.txt")
```

▶ # 각 행마다 행 번호 부여

```
write.table(iris,"iris2.txt", row.name=F,sep="\t")
```

▶ # 행 번호 삭제,
구분기호로 탭 사용

```
write.table(iris,"iris3.txt", row.name=F,sep=",")
```

▶ # 행 번호 삭제,
구분기호로 콤마 사용



03

데이터 구조

- 1) R의 데이터 구조
- 2) 벡터
- 3) 행렬
- 4) 배열
- 5) 데이터 프레임
- 6) 리스트
- 7) 공공 데이터 분석

1) R의 데이터 구조

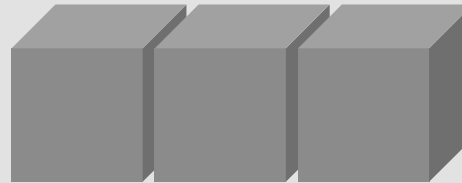
벡터 (Vector)

행렬 (Matrix)

배열 (Array)

데이터프레임
(Data Frame)

리스트 (List)



1) R의 데이터 구조

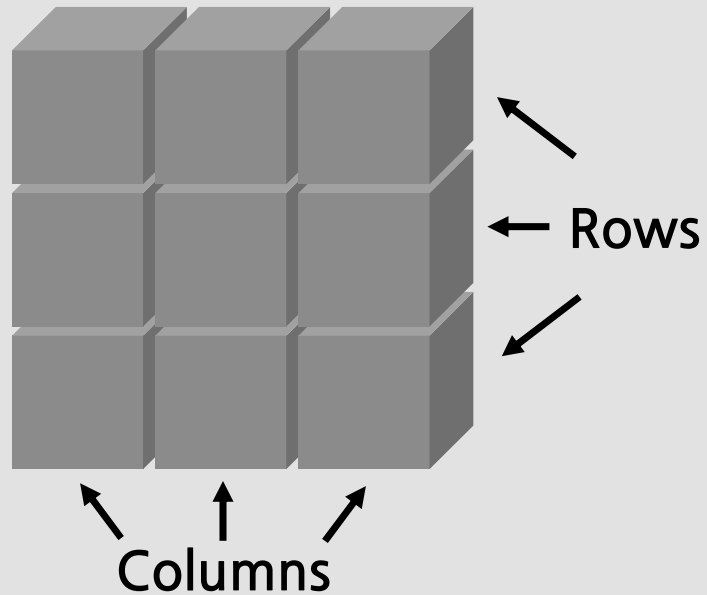
벡터 (Vector)

행렬 (Matrix)

배열 (Array)

데이터프레임
(Data Frame)

리스트 (List)



1) R의 데이터 구조

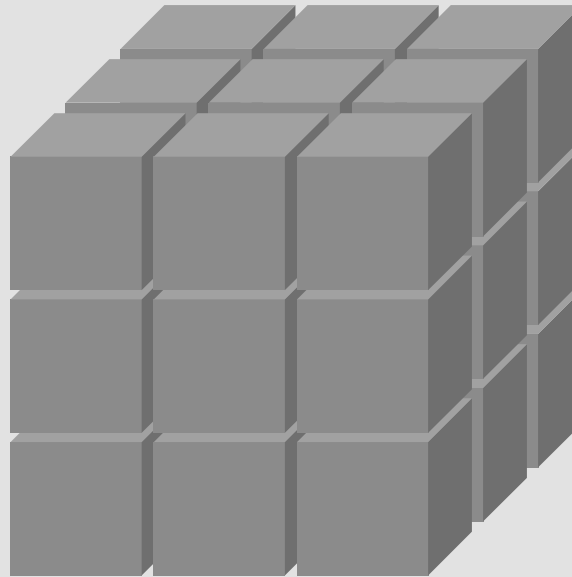
벡터 (Vector)

행렬 (Matrix)

배열 (Array)

데이터프레임
(Data Frame)

리스트 (List)



1) R의 데이터 구조

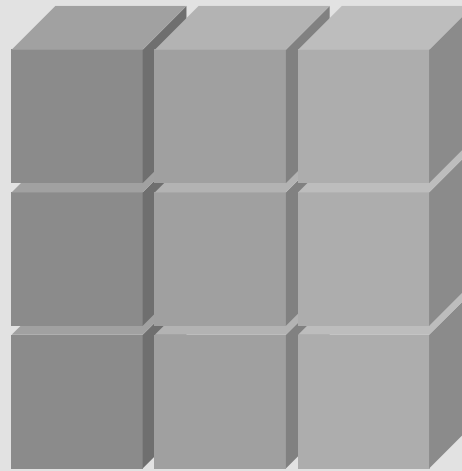
벡터 (Vector)

행렬 (Matrix)

배열 (Array)

**데이터프레임
(Data Frame)**

리스트 (List)



(Table)

1) R의 데이터 구조

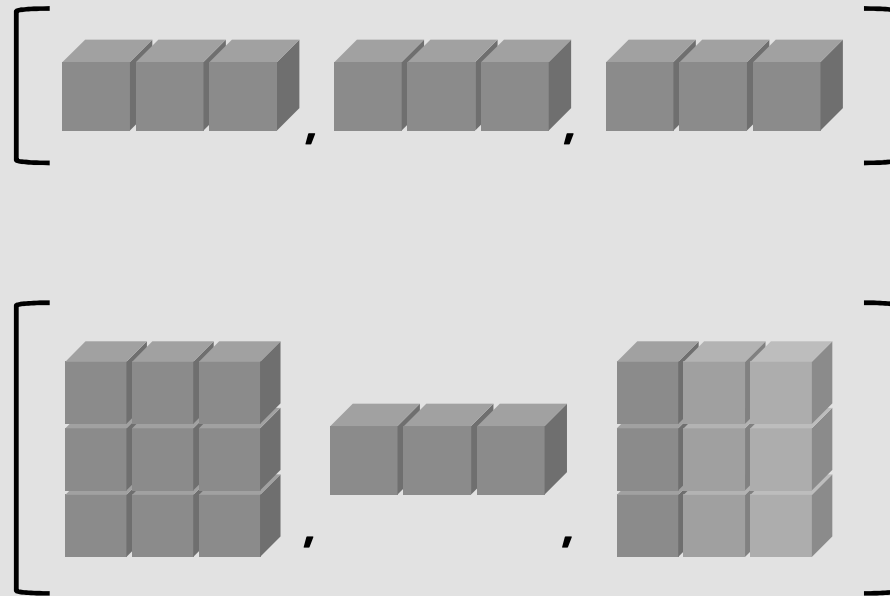
벡터 (Vector)

행렬 (Matrix)

배열 (Array)

데이터프레임
(Data Frame)

리스트 (List)



2) 벡터

벡터(Vector)

R의 가장 기본이 되는 자료 객체로 1차원 데이터 구조



1개 이상의 원소로 구성된 자료 구조



한 가지 형태만 입력이 가능함

벡터를 만드는 방법

`c()` , `rep()`, `seq()`

2) 벡터

```
v1 <- c(1, 2, 3, 4, 5)
```

▶ # v1에 1, 2, 3, 4, 5로 구성된 벡터를 입력함

```
v1
```

```
v2 <- c(1:5)
```

▶ # v2에 1, 2, 3, 4, 5로 구성된 벡터를 입력함

```
v2
```

```
rep(c(1,2), times=2)
```

▶ # 1과 2로 구성된 벡터를 2회 반복함

2) 벡터

```
rep(c(2,4), length=6)
```

▶ # 2와 4로 구성된 벡터를 반복하되 크기는 6으로 지정함

```
seq(1, 10, by=2)
```

▶ # 1에서 10까지 2씩 증가되는 벡터를 작성함

```
x <- c(10,35,20,80,90)
```

▶ # x에 벡터를 입력함

```
sum(x)
```

▶ # x의 합계를 구함

```
mean(x)
```

▶ # x의 평균을 구함

3) 행렬

행렬(Matrix)

동일한 형태로 구성된 2차원 데이터 구조



행과 열로 이루어짐

행렬을 만드는 방법

`matrix()`

```
matrix(data = NA, nrow = 1, ncol = 1,  
       byrow = FALSE, dimnames = NULL)
```


3) 행렬

```
m1 <- matrix(1:12, nrow=3) ► # 3행 4열 행렬
```

```
m1
```

```
m1[2,3] ► # 2행 3열의 원소 출력
```

3) 행렬

```
m2 <- matrix(1:9, ncol=3) ▶ # 3행 3열 행렬
```

```
m2
```

```
m3 <- matrix(1:9, nrow=3, dimnames =  
list(c("r1","r2","r3"), c("c1","c2","c3")))
```

```
▶ # 행과 열에 이름 지정
```

```
m3
```

4) 배열

배열(Array)

동일한 형으로 구성된 2차원 이상의 데이터 구조

배열 만드는 방법

array()

```
array(data = NA, dim=c(행 개수, 열 개수,  
  행렬 개수, ...), dim.names = NULL)
```

4) 배열

```
array(1:6)
```

▶ # 1차원 배열 생성

```
array(1:6,c(2,3))
```

▶ # 2차원 배열 생성

```
array(1:8,c(2,2,2))
```

▶ # 3차원 배열 생성

5) 데이터 프레임

데이터 프레임(Data Frame)

데이터베이스에서 테이블과 유사한 데이터 객체



각 열들이 서로 다른 형태의 객체를 가질 수 있음

데이터 프레임 만드는 방법

`data.frame()`

```
data.frame(객체1, 객체2, ...)
```

5) 데이터 프레임

```
name <- c("LEE", "KIM", "BONG", "PARK")
```

```
kor <- c(100, 50, 88, 75)
```

```
eng <- c(95, 68, 90, 100)
```

```
mat <- c(100, 45, 75, 80)
```

```
data.frame(name, kor, eng, mat)
```

▶ # name, kor, eng, mat로 데이터 프레임 생성

6) 리스트

리스트(List)

서로 다른 유형의 데이터 구조 결합

리스트 만드는 방법

`list()`

6) 리스트

```
v <- c(1:6)
```

▶ # 벡터 생성

```
m <- matrix(c(1:12), nrow=3)
```

▶ # 행렬 생성


```
l <- list(v,m)
```

▶ # 리스트 생성

```
l
```


7) 공공 데이터 분석

- 공공 데이터 읽어서 확인하기

- 공공데이터포털(<https://www.data.go.kr/>) 접속
 - '기장 인구' 검색
 - 다운로드 후 필드명을 영문으로 변경
 - 숫자의 콤마(,)를 없앰
- 



7) 공공 데이터 분석

```
pop_gijang <- read.csv("gijang.csv")
```

▶ # gijang.csv를 읽어옴

```
pop_gijang
```

```
head(pop_gijang)
```

▶ # 상위 6개의 데이터 확인

```
tail(pop_gijang)
```

▶ # 하위 6개의 데이터 확인

7) 공공 데이터 분석

```
class(pop_gijang)
```

▶ # 데이터 프레임 구조임을 확인함

```
str(pop_gijang)
```

▶ # 행, 변수, 각 변수의 구조를 확인

```
summary(pop_gijang)
```

▶ # 데이터 요약

학습 평가

Q1

Q2

Q1

다음 중 R에서 명령어에 대한 도움말을
보기 위한 함수에 해당하는 것은?

1 example()

3 q()

2 help()

4 manual()

학습 평가

Q1

Q2

Q1

다음 중 R에서 명령어에 대한 도움말을
보기 위한 함수에 해당하는 것은?

1 example()

3 q()

☒ 2 help()

4 manual()

정답

2 help()

해설

R에서 도움말을 보기 위해서는 ? 또는 help()를
사용합니다.

학습 평가

Q1

Q2

Q2

다음 명령문을 실행한 결과로 옳은 것은?

```
rep(c(2,4), times=3)
```

- 1 2 4 2 4 2 4
- 2 2 3 4 2 3 4
- 3 2 4
- 4 2 3 4 2 3 4

학습 평가

Q1

Q2

Q2

다음 명령문을 실행한 결과로 옳은 것은?

```
rep(c(2,4), times=3)
```



2 4 2 4 2 4

2

2 3 4 2 3 4

3

2 4

4

2 3 4 2 3 4 2 3 4

정답

1

2 4 2 4 2 4

해설

2 4로 구성된 벡터를 3회 반복합니다.

정리 하기

R 프로그램 개요

- ✓ R 프로그램의 개요
 - R 프로그램은 로스 이하카와 로버트 젠틀만에 의해 개발됨
 - R 프로그램은 통계, 머신 러닝, 금융, 시각화 등의 다양한 패키지를 갖추고 있으며 무료로 제공됨
 - R 프로그램은 오픈 소스로 수많은 R 사용자들이 자유롭게 분석 기법을 추가하는 것이 가능함

정리 하기

R 프로그램 개요

- ✓ R 프로그램 설치
 - R은 CRAN(The Comprehensive R Archive Network)을 통해 최신 버전을 다운로드 할 수 있음
 - R Studio는 R을 편리하게 사용할 수 있는 사용자 인터페이스를 제공하는 프로그램

정리 하기

데이터의 입력과 출력

- ✓ 데이터의 입력
 - `c()` 함수를 이용함
 - `read.table()` 함수를 이용함
 - `read.csv()` 함수를 이용함
- ✓ 데이터의 출력
 - `write.table()` 함수를 이용함

정리 하기

데이터 구조

- ✓ 벡터
 - R의 가장 기본이 되는 자료 객체로 1차원 데이터 구조
- ✓ 행렬
 - 동일한 형태로 구성된 2차원 데이터 구조
- ✓ 배열
 - 동일한 형으로 구성된 2차원 이상의 데이터 구조

정리 하기

데이터 구조

- ✓ 데이터 프레임
 - 데이터베이스에서 테이블과 유사한 데이터 객체
- ✓ 리스트
 - 서로 다른 유형의 데이터 구조 결합

The background is a dark, abstract composition featuring a network of glowing white lines. A prominent hexagon is located in the upper right quadrant, with lines radiating from it. On the left side, there are several overlapping, parallel lines that create a sense of depth and movement. The overall aesthetic is futuristic and technological.

- 다음 시간에 살펴 볼 내용 -

06강 빅데이터 시각화

수고하셨습니다.