

R컴퓨팅

1강

R입문

정보통계학과 장영재 교수

- 1 R 프로그램 소개
- 2 R 프로그램 설치
- 3 R 프로그램 실행
- 4 R의 기본 활용
- 5 R 패키지의 활용
- 6 R에서 도움말 찾기

1

R 프로그램 소개

1 R 프로그램 소개

- R 프로그램은 데이터분석과 그래프 작성 등을 위하여 개발된 오픈 소스 데이터분석용 프로그램
- R 프로그램은 뉴질랜드 오클랜드 대학의 Robert Gentleman 과 Ross Ihaka 에 의해 개발

현재 통계학 및 컴퓨터 과학 분야 등의 학자들로 이루어진
R Development Core Team 에 의하여 지속적으로 유지되고 개선

1 R 프로그램 소개

▶ R 프로그램이 가지고 있는 특징을 요약하면 다음과 같음

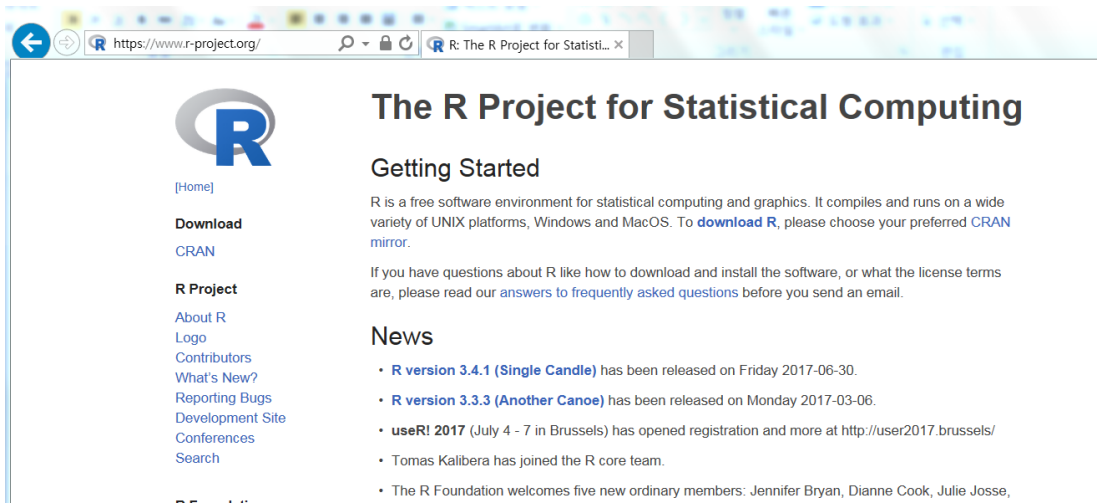
- 무료프로그램
- 오픈소스이기 때문에 수많은 R 사용자들이 자유롭게 분석기법들을 추가할 수 있음
- 다양한 분야의 통계분석이 가능하고,
특히 그래픽 기능이 다른 통계패키지에 비하여 뛰어남
- 도움말 기능이 뛰어남

2 R 프로그램 설치

2


R 프로그램 설치

- R 프로그램은 홈페이지인 CRAN (Comprehensive R Archive Network) 사이트를 통해 최신 버전을 다운로드 할 수 있음
- CRAN 사이트의 주소는 <http://www.r-project.org> 이며, 보다 빠른 다운로드를 위해서는 세계 여러 나라의 미러(Mirror) 사이트를 이용 권장



2

R 프로그램 설치



CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: [windows release](#), [windows old release](#).

If you want to host a new mirror at your institution, please have a look at the [CRAN Mirror HOWTO](#).

0-Cloud	
https://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio
http://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio
Algeria	
https://cran.usthb.dz/	University of Science and Technology Houari Boumediene
http://cran.usthb.dz/	University of Science and Technology Houari Boumediene
Argentina	
http://mirror.fcaglp.unlp.edu.ar/CRAN/	Universidad Nacional de La Plata
Australia	
https://cran.csiro.au/	CSIRO
http://cran.csiro.au/	CSIRO
https://mirror.aarnet.edu.au/pub/CRAN/	AARNET
https://cran.curtin.edu.au/	Curtin University of Technology
Austria	
https://cran.wu.ac.at/	Wirtschaftsuniversität Wien
http://dssm.unipa.it/CRAN/	Università degli Studi di Palermo
Japan	
https://cran.ism.ac.jp/	The Institute of Statistical Mathematics, Tokyo
http://cran.ism.ac.jp/	The Institute of Statistical Mathematics, Tokyo
Korea	
http://cran.nexr.com/	NexR Corporation, Seoul
http://healthstat.snu.ac.kr/CRAN/	Graduate School of Public Health, Seoul National University, Seoul
http://cran.biodisk.org/	The Genome Institute of UNIST (Ulsan National Institute of Science and Technology)

※ R 프로그램 홈페이지인 CRAN 사이트

3 R 프로그램 실행

3 R 프로그램 실행

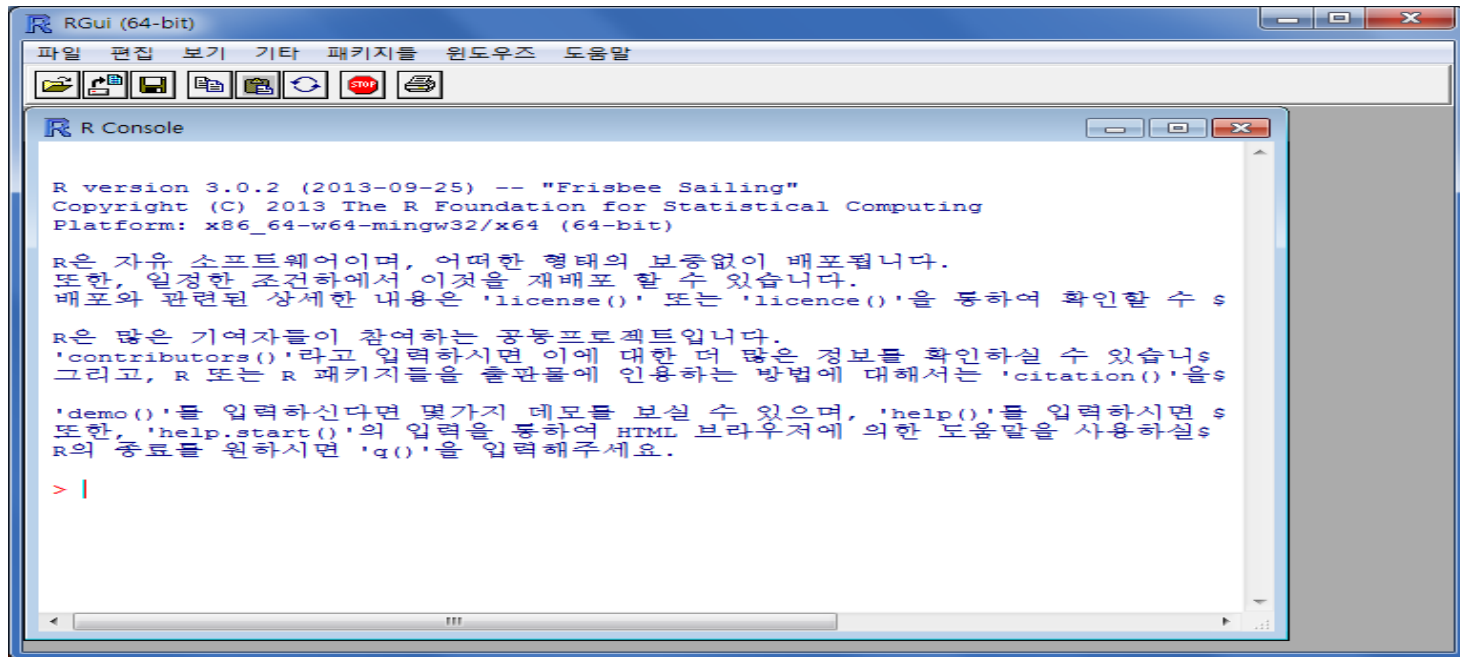
- R 프로그램은 바탕화면에 생성된 아이콘을 더블 클릭하면 실행할 수 있으며 시작 → 모든 프로그램 → R → R 3.4.1을 선택하는 방법도 있음
- 초기화면이 나타나고 R이 시작되면 > 기호가 나타나는데, 이것이 R 프로그램의 프롬프트임

R 프로그램은 대화식으로서, R 프롬프트 다음에 명령문을 입력하고 Enter 키를 치면 입력된 명령문이 실행

3

R 프로그램 실행

※ R 프로그램 초기화면



3 R 프로그램 실행

1 명령문 입력

➤ 보기 1-1: 1과 2를 더하는 연산의 수행

```
> 1+2  
[1] 3
```

R 프로그램에서 연산의 결과는 벡터로 인식하는데
정답3 앞에 있는 [1]는 벡터의 첫 번째 원소를 의미

3 R 프로그램 실행

1 명령문 입력

- 보기 1-2: min()함수를 이용하여 (1, 3, 5, 7, 9)로 이루어진 벡터의 원소 중 최솟값 구하기

```
> min(1,3,5,7,9)  
[1] 1
```

- R 프로그램에서 명령문은 여러 줄에 걸쳐 표현할 수도 있는데 이 경우에 R 프롬프트는 > 이 아니라 + 로 변경

```
> min(1,3,5,  
+ 7,9)  
[1] 1
```

3 R 프로그램 실행

1 명령문 입력

- ▶ 만약 명령문을 입력할 때 오류가 있다면 R 프로그램에서는 오류의 이유를 출력하므로, 이를 쉽게 수정할 수 있음

```
> man(1,3,5,7,9)
```

에러: 함수 "man"를 찾을 수 없습니다

- ▶ 이때 키보드의 위쪽 화살표를 누르면, 직전에 입력했던 명령문이 나타남

3 R 프로그램 실행

1 명령문 입력

➤ 보기 1-3: 보기 1-1과 보기 1-2의 명령을 한 줄에 입력하여 수행

```
> 1+2; min(1,3,5,7,9)  
[1] 3  
[1] 1
```

세미콜론은 명령문과 명령문을 구분 짓는 역할

3 R 프로그램 실행

2 스크립트 실행

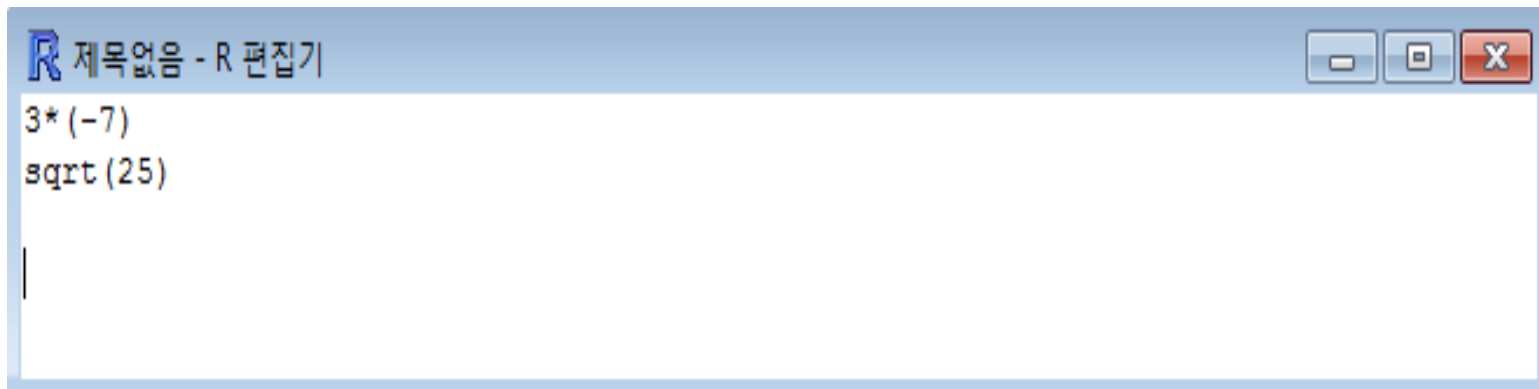
- R Console 창에서 필요한 명령문을 입력해서 결과를 얻는 방식은 대화식이지만 만약 명령문의 개수가 많거나 수정이 필요한 경우 편집기를 활용
- 우선 수행하고자 하는 명령문을 미리 기록한 후 한꺼번에 실행하는 방법도 사용 가능

R 편집기는 R Console 풀다운 메뉴의 '파일' → '새 스크립트'로 실행

3 R 프로그램 실행

2 스크립트 실행

- 보기 1-4: 다음과 같은 스크립트를 입력한 후, R 편집기 풀다운 메뉴의 '편집' → '전부실행'을 클릭하여 작업수행



The screenshot shows a window titled '제목없음 - R 편집기' (Untitled - R Editor). The window contains a text area with the following R code:

```
3*(-7)
sqrt(25)
```

The code is entered on two separate lines. The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

- R Console 창에 나타나는 작업결과는 다음과 같음

3 R 프로그램 실행

2 스크립트 실행

```
> 3*(-7)  
[1] -21  
> sqrt(25)  
[1] 5
```

- ▶ 일부만을 실행시키고자 할 때에는, 원하는 부분만 영역으로 선택한 후 R 편집기 풀다운 메뉴의 '편집' → '줄 또는 선택영역 실행'을 클릭

3 R 프로그램 실행

2 스크립트 실행

- R 편집기에 입력된 명령문들은 R 편집기 풀다운 메뉴의 ‘파일’ → ‘다른 이름으로 저장’을 선택하면 별도의 파일로 저장할 수 있음
- R 편집기에서 기본으로 사용하고 있는 확장자는 ‘.R’

3 R 프로그램 실행

2 스크립트 실행

- R 프로그램을 종료하기 위해서는 R 프롬프트에서 "q()" 명령문을 입력하거나, 풀다운 메뉴의 '파일' → '종료'를 선택
- 계산 시간이 막대한 R 명령어를 입력하여 실행하는 도중에 이를 중지하고 다시 R 프롬프트 상태로 돌아오고자 한다면, Esc 키를 누르거나 빨강색의 stop 버튼을 클릭
 - 이 경우에 R 프로그램은 종료되지 않고, R 프롬프트 상태에서 명령문을 다시 입력할 수 있게 됨
- R 프로그램 종료 시 작업공간 저장여부를 묻게 되는데 아니오를 선택하면 작업공간이 저장되지 않음

4 R의 기본 활용

4 R의 기본 활용

1 변수의 활용

- ▶ 변수를 이용해서 작업을 하는 것이 길고 복잡한 많은 계산을 요하는 경우에는 훨씬 수월
- ▶ R에서 변수에 값을 할당하는 방법은 '=' 혹은 '<-' 기호를 이용

4 R의 기본 활용

1 변수의 활용

- ▶ 보기 1-5: 아래 예는 R에서 변수를 할당하고 연산을 수행한 뒤 그 값을 출력하는 절차

```
> x = 5  
> y = -3  
> z = sqrt(x^3 + y^2)  
> z  
[1] 11.57584  
> print(z)  
[1] 11.57584
```

- ▶ R에서 사용 가능한 변수의 이름은 다음의 규칙을 따라야만 함

4 R의 기본 활용

1 변수의 활용

- 문자 a ~ z, A ~ Z, 숫자 0 ~ 9, 그리고 '.'과 '_'의 조합으로 구성
- 이름의 첫 글자로 숫자와 '_'은 사용할 수 없음
- 대문자와 소문자는 서로 구분

4 R의 기본 활용

1 변수의 활용

※R에서 사용 가능한 주요 수학 함수

함수	설명	사용 예
abs(x)	절대값	> abs(-5) [1] 5
ceiling(x)	x보다 같거나 큰 값 중 가장 적은 정수값	> ceiling(2.35) [1] 3
choose(x, y)	x개에서 y개를 고르는 조합의 경우의 수	> choose(5,3) [1] 10
cos(x)	코사인 삼각함수값	> cos(1) [1] 0.5403023
exp(x)	지수함수 e^x 값	> exp(-3) [1] 0.04978707

4 R의 기본 활용

1 변수의 활용

※R에서 사용 가능한 주요 수학 함수

함수	설명	사용 예
factorial(x)	x의 계승 $x!$ 값	> factorial(5) [1] 120
floor(x)	x보다 같거나 작은 값중 가장 큰 정수값	> floor(2.35) [1] 2
log(x)	x의 자연로그값	> log(10) [1] 2.302585
log10(x)	x의 사용로그값	> log10(10) [1] 1
log2(x)	밑이 2인 x의 로그값	> log2(4) [1] 2

4 R의 기본 활용

1 변수의 활용

※R에서 사용 가능한 주요 수학 함수

함수	설명	사용 예
pi	원주율 파이값	<pre>> pi [1] 3.141593</pre>
round(x, digits=n)	x를 소수점이하 n자리로 반올림	<pre>> round(2.34567, 2) [1] 2.35</pre>
signif(x, digits=n)	x를 유효수 n자리로 반올림	<pre>> signif(0.0023456, 2) [1] 0.0023</pre>
sin(x)	사인 삼각함수 값	<pre>> sin(1) [1] 0.841471</pre>
sqrt(x)	x의 제곱근 함수	<pre>> sqrt(9) [1] 3</pre>

4 R의 기본 활용

1 변수의 활용

※R에서 사용 가능한 주요 수학 함수

함수	설명	사용 예
<code>tan(x)</code>	탄젠트 삼각함수 값	<pre>> tan(1) [1] 1.557408</pre>
<code>trunc(x)</code>	x의 소수점 이하 버림	<pre>> trunc(3.141593) [1] 3</pre>

4 R의 기본 활용

1 변수의 활용

- R 작업 중 생성된 변수의 리스트를 'ls()' 명령문을 이용하여 출력시킬 수 있으며 'rm()' 명령문으로 생성된 변수를 지울 수 있음

```
> ls()
```

```
[1] "x" "y" "z"
```

```
> rm(x) # 변수 'x' 삭제
```

```
> ls()
```

```
[1] "y" "z"
```

```
> rm(list=ls()) # 모든 변수 삭제
```

```
> ls()
```

```
character(0)
```

4 R의 기본 활용

1 변수의 활용

➤ '`rm(list=ls())`' 명령문은 작업 중에 생성된 모든 변수를 삭제

4 R의 기본 활용

2 벡터의 생성

- 보기 1-6: c() 함수를 통해 (3,4,5), (9,16,25)의 값을 갖는 벡터 v1, v2를 만들고 두 벡터를 병합하여 v3 벡터를 생성

```
> v1 = c(3,4,5)
```

```
# 벡터 'v1' 생성
```

```
> v1
```

```
[1] 3 4 5
```

```
> v2 = c(9,16,25)
```

```
# 벡터 'v2' 생성
```

```
> v2
```

```
[1] 9 16 25
```

```
> v3 = c(v1,v2)
```

```
# 벡터 'v3' 생성
```

```
> v3
```

```
[1] 3 4 5 9 16 25
```

4 R의 기본 활용

2 벡터의 생성

- 보기 1-7: c() 함수를 통해 (a,b,c), (0,1,2)의 값을 갖는 벡터 s1, s2를 만들고 이를 병합하여 s3라는 벡터를 생성

```
> s1 = c("a","b","c") # 문자형 벡터 's1' 생성
```

```
> s1
```

```
[1] "a" "b" "c"
```

```
> s2 = c(0,1,2) # 숫자형 벡터 's2' 생성
```

```
> s2
```

```
[1] 0 1 2
```

```
> s3 = c(s1,s2)
```

```
> s3
```

```
[1] "a" "b" "c" "0" "1" "2"
```

```
> mode(s3) # 벡터의 형태를 출력
```

```
[1] "character"
```

4 R의 기본 활용

3 벡터의 통계값 계산

- 평균이나 표준편차와 같은 기본적인 통계값을 벡터에 대하여 계산
- ※ 주요 통계함수

함수	설명	사용 예
IQR(x)	벡터 x의 사분위 범위	<pre>> v1 = c(1, 3, 5, 7, 9, 11, 13, 15, 17, 19) > IQR(v1) [1] 9</pre>
max(x)	벡터 x의 최댓값	<pre>> max(v1) [1] 19</pre>
mean(x)	벡터 x의 평균값	<pre>> mean(v1) [1] 10</pre>
median(x)	벡터 x의 중앙값	<pre>> median(v1) [1] 10</pre>

4 R의 기본 활용

3 벡터의 통계값 계산

※ 주요 통계함수

함수	설명	사용 예
min(x)	벡터 x의 최소값	<pre>> min(v1) [1] 1</pre>
quantile(x, q)	벡터 x의 q 백분위 수	<pre>> quantile(v1, 0.05) 5% 1.9</pre>
range(x) diff(range(x))	벡터 x의 최소값과 최댓값 벡터 x의 범위	<pre>> range(v1) [1] 1 19 > diff(range(v1)) [1] 18</pre>
sd(x)	벡터 x의 표준편차값	<pre>> sd(v1) [1] 6.055301</pre>

4 R의 기본 활용

3 벡터의 통계값 계산

※ 주요 통계함수

함수	설명	사용 예
sum(x)	벡터 x의 총합	<pre>> sum(v1) [1] 100</pre>
var(x)	벡터 x의 분산값	<pre>> var(v1) [1] 36.66667</pre>

➤ 결측치가 NA로 표현되는데, 만약 벡터의 원소 중에 NA가 포함되어 있다면 통계계산 함수의 사용에 주의

4 R의 기본 활용

3 벡터의 통계값 계산

- ▶ 보기 1-8: (1, 3, 5, 7, 9, 11, 13, 15, 17, 19, NA)로 이루어진 벡터 v1을 정의하고 mean() 함수를 실행

```
> v1 = c(1, 3, 5, 7, 9, 11, 13, 15, 17, 19, NA)
> mean(v1)
[1] NA
```

- ▶ 통계함수의 계산에서 NA값을 포함시키지 않고 계산하도록 하려면 다음과 같이 na.rm=TRUE 옵션을 사용

```
> mean(v1,na.rm=TRUE)
[1] 10
```

4 R의 기본 활용

3 벡터의 통계값 계산

- 한 벡터 내에 결측치가 있는지 불확실한 경우에는 `is.na()` 함수를 사용하면 확인

```
> v2 = c(1, 3, 5, 7, 9, NA)
```

```
> is.na(v2)
```

```
[1] FALSE FALSE FALSE FALSE FALSE TRUE
```

4 R의 기본 활용

4 R 작업 폴더의 설정

- R 프로그램 실행 후 만들어질 수 있는 여러 개의 파일을 하나의 공통폴더에 저장하는 것이 추후 파일의 관리에 용이
 - 풀다운 메뉴의 '파일' → '작업 디렉토리 변경'을 이용하거나 R console 에서 `setwd()` 함수를 이용해서 작업폴더를 변경
 - 이미 설정된 작업폴더는 함수 `getwd()`를 통해 확인
-
- `> setwd("D:/R files")` # 작업폴더의 경로에 “\”(back slash) 이 아닌 # “/”(slash) 가 사용되었음에 유의
 - `# setwd("D:\\\\R files")`와 같이 “\\” 사용 가능
 - `> getwd()`
 - `[1] "D:/R files"`

4 R의 기본 활용

4 R 작업 폴더의 설정

- 보기 1-9: 현재 R 편집기에 입력된 명령문들을 “D:/R files/example1.R” 이라는 이름으로 저장
- R 편집기 풀다운 메뉴의 ‘파일’ → ‘다른 이름으로 저장’을 선택하면 “D:/R files” 경로가 자동 선택되고 파일이름에 “example1” 이라 입력하면 “D:/R files/example1.R” 이라는 파일이 생성

4 R의 기본 활용

5 R 작업기록의 저장 및 불러오기

- R 프로그램을 시작한 이후 지금까지 작업했던 모든 명령문 내역을 저장할 수 있음
- R console 풀다운 메뉴의 '파일' → '작업기록 저장하기'를 선택하면 원하는 파일이름으로 저장할 수 있으며 풀다운 메뉴의 '파일' → '작업기록 불러오기'를 이용하면 불러올 수 있음

- 기본으로 사용하고 있는 확장자는 '.Rhistory'

4 R의 기본 활용

5 R 작업기록의 저장 및 불러오기

- 보기 1-10: 1.4.4절에서 저장해 두었던 R의 명령문 파일인 “D:/R files/example1.R” 파일의 내용을 R에서 한 번에 실행시켜 보도록 하자.

```
> setwd("D:/R files")  
> source("example1.R",echo=TRUE)  
# 중간생략 #
```

```
> v4[v4 < median(v4)]  
[1] 1 1 2 2 3 3 4 4 5 5 1 1 2 2 3 3 4 4 5 5  
> v4[ v4 <= quantile(v4, 0.05) ]  
[1] 1 1 1 1
```


4 R의 기본 활용

5 R 작업기록의 저장 및 불러오기

- R에서는 `source()` 함수를 이용하고 `echo=TRUE`라는 옵션을 사용하면 외부파일로 저장된 R 명령문 전체 내용을 한 번에 실행시킬 수도 있음

4 R의 기본 활용

6 작업공간

- R 에서 작업공간(workspace)이란 R 실행과정을 통해 생성된 벡터 등과 같은 객체와 사용자 정의 함수 등을 확장자 '.RData' 인 파일로 저장
- R 프로그램을 새로 시작하였을 때, 이전 작업에서 생성했던 복잡한 작업내용들이 그대로 복원되는 효과
- 작업공간을 별도의 파일로 저장하고자 할 때에는 R Console 풀다운 메뉴의 '파일' → '작업공간 저장하기'나 R Console상 명령문으로 실행

4 R의 기본 활용

6 작업공간

```
> save.image(file="example1.RData")  
> load(file="example1.RData")
```

- `save.image()` 함수는 현재 실행된 모든 객체를 저장하는 것이고, `load()` 함수는 이미 파일로 저장된 객체를 불러내어 활성화
- R 프로그램을 종료할 때 작업공간 이미지 저장여부 질문에 “예”를 선택하면, 작업공간은 자동으로 작업폴더 내 “.RData”라는 파일로 자동 저장

5 R패키지의 활용

5 R 패키지의 활용

1 패키지 소개

- R에서 패키지란 특정 분석을 수행할 수 있는 함수, 객체, 도움말, 데이터 등의 집합
- 예를 들어 `mean()`과 같은 기본 통계함수를 모은 패키지의 이름은 `stats`이며 패키지들을 확인해 보는 방법은 `library()`
- 다음은 R을 설치했을 때 자동적으로 설치되는 패키지의 목록 중 일부인데 이를 확인하려면 '`search()`' 명령문을 사용

```
> search()
```

```
[1] ".GlobalEnv" "package:stats" "package:graphics"
```

```
[4] "package:grDevices" "package:utils" "package:datasets"
```

```
[7] "package:methods" "Autoloads" "package:base"
```

5 R 패키지의 활용

1 패키지 소개

- R을 설치할 때 자동적으로 설치되는 되지만, 사용자가 직접 활성화 시켜야만 하는 recommended 패키지도 있음
- 활성화 시키는 방법은 `library(패키지명)`을 명령문으로 사용
 - 예를 들어 MASS 패키지를 활성화 시키는 방법은 다음과 같음

```
> library(MASS)
> help(package="MASS")
starting httpd help server ... 완료
> detach(package:MASS)
```

5 R 패키지의 활용

1 패키지 소개

- 위에서 `help(package="MASS")` 명령문은 MASS 패키지에 대한 자세한 설명창을 열며, `detach(package:MASS)` 명령문은 불러온 패키지의 활성화를 해제
- `data(package="패키지명")` 명령문을 사용하면 설치되어 있는 패키지내에 저장되어 있는 데이터를 확인하고 불러오는 것도 가능
 - `head()` 명령문은 데이터의 초반부만 보여주는 명령문
 - > `data(package="MASS")`
 - > `data(Boston, package="MASS")`
 - > `head(Boston)`

5 R 패키지의 활용

2 패키지의 설치 및 활용

- 인터넷 사이트 '<http://cran.r-project.org/web/views/>'를 방문하여 보면, 통계 분야별로 사용 가능한 패키지를 분류하여 소개하고 있음
- '<http://www.rseek.org/>' 사이트에서는 키워드를 이용하여 패키지를 탐색하는 방법도 제공
 - 한국의 R 사용자 모임 : '<http://www.r-project.kr>'도 유용

5 R 패키지의 활용

2 패키지의 설치 및 활용

- R Console 풀다운 메뉴의 '패키지들' → '패키지(들) 설치하기'를 선택하면 미러사이트 목록이 나타나는데 사용자의 위치와 가까운 미러사이트를 선택하면 설치 가능한 패키지의 목록이 나타남
- 이 화면에서 설치하고자 하는 패키지를 선택하면 설치과정이 진행

6 R에서 도움말 찾기

6

R에서 도움말 찾기

- R을 사용하다 보면, 활용하고자 하는 함수에 대한 도움말을 얻고자 할 때 `help`나 `?` 명령어를 사용

```
> help(mean)
```

```
> ?mean
```

```
> example(mean)
```

```
mean> x <- c(0:10, 50)
```

```
mean> xm <- mean(x)
```

```
mean> c(xm, mean(x, trim = 0.10))
```

```
[1] 8.75 5.50
```

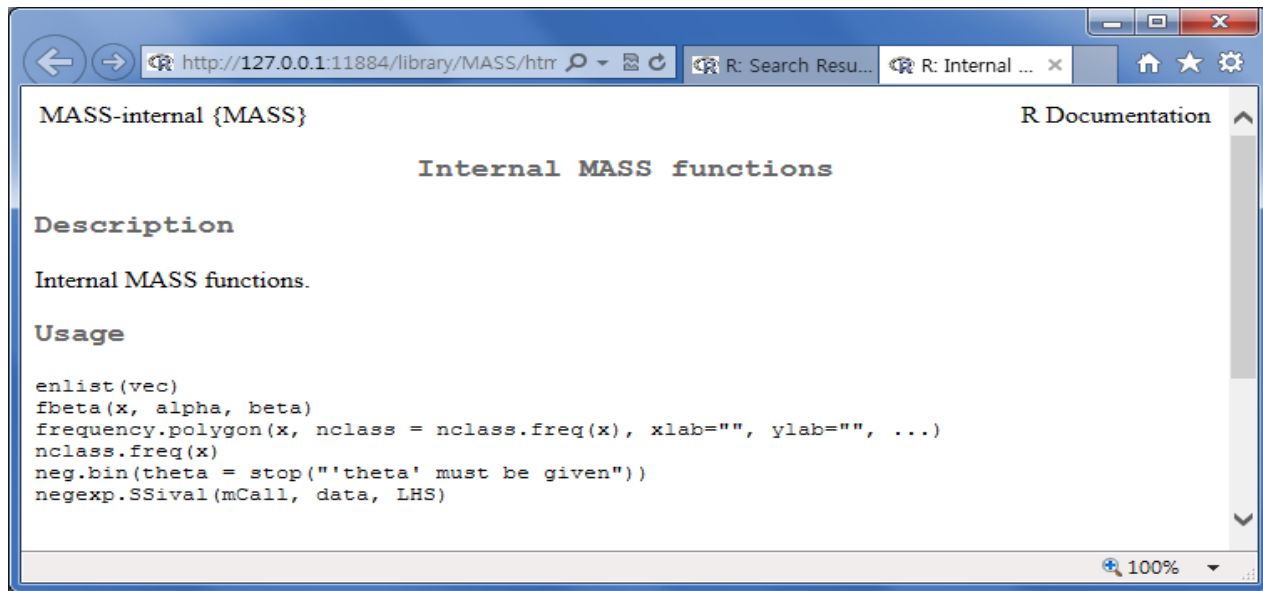
- 정확한 함수이름은 모르지만 특정단어를 포함하는 함수를 검색하고자 할 때에는 `help.search()` 명령문을 사용
(이미 설치되어 있는 패키지의 함수만을 검색)

6

R에서 도움말 찾기

- MASS 패키지 내의 enlist 함수에 대한 도움말을 찾고자 한다면, 해당되는 링크를 클릭하거나, `help(enlist, package="MASS")` 라고 입력

※ `help(enlist, package="MASS")` 명령문의 결과



6

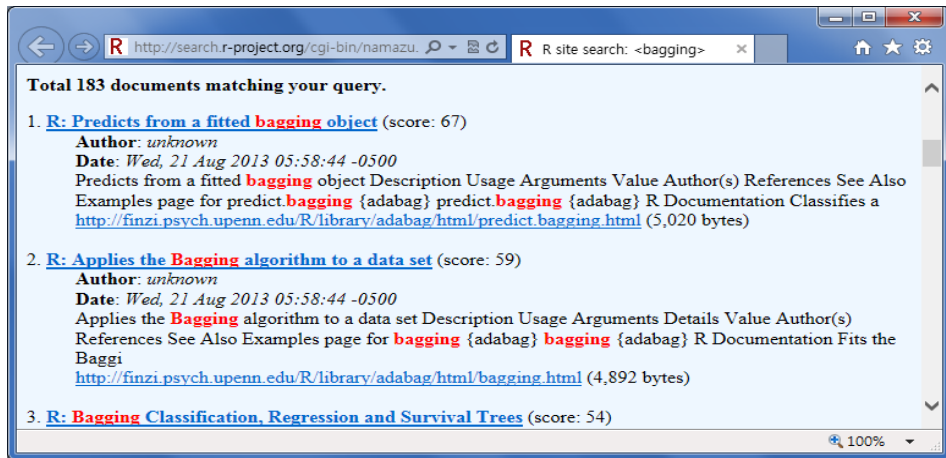
R에서 도움말 찾기

- ▶ 만약 아직 설치하지 않은 패키지의 함수를 검색하고자 한다면, RSiteSearch() 명령문을 사용

› RSiteSearch("bagging")

서치쿼리가 <http://search.r-project.org>에 전달되었습니다
곧 브라우저에서 결과를 확인할 수 있습니다

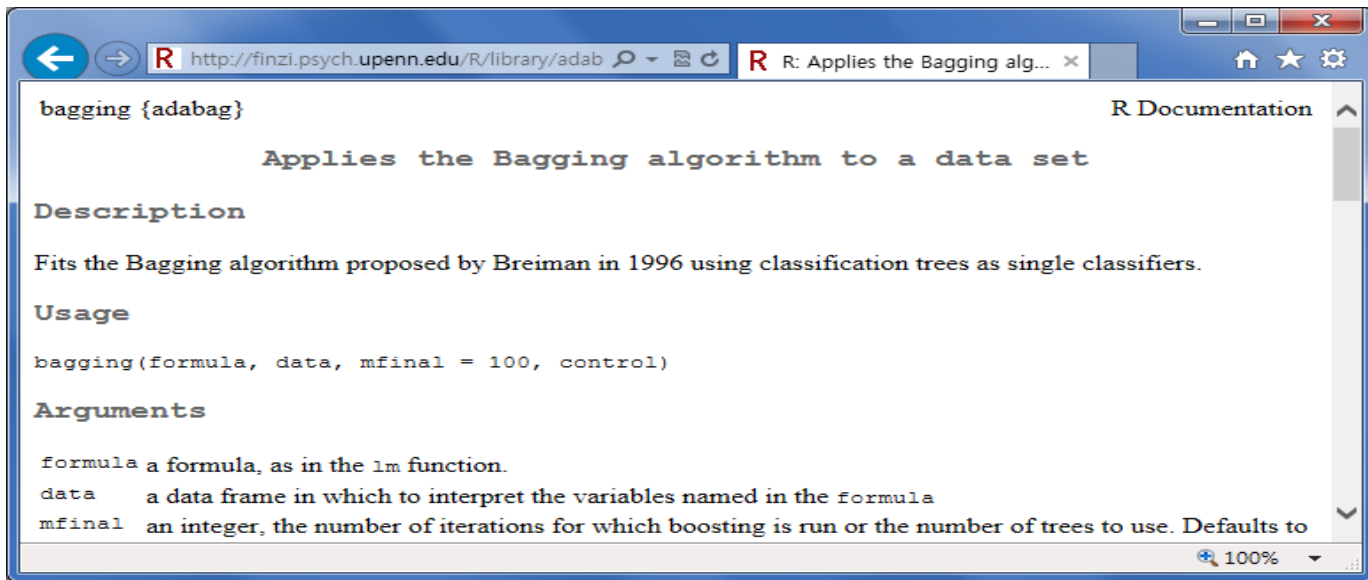
※ 'RSiteSearch("bagging")' 명령문의 결과



6

R에서 도움말 찾기

※ bagging 함수의 도움말



- bagging {adabag}이란 표현은 bagging()함수를
사용하려면 adabag이라는 패키지를 설치하여야 함을 의미

R컴퓨팅

