

Introduction to R

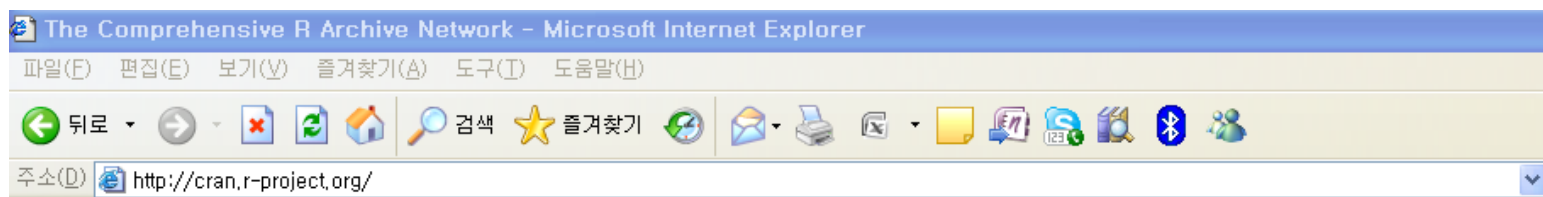
Sungsu Lim
Applied Algorithm Lab.
KAIST

R의 소개

- R은 S와 S-PLUS의 환경을 기초로 해서 만들어진 통계적 도구이다. (1995년 Robert Gentleman, Ross Ihaka 개발)
- R은 무료이고 공개되어 있으며 Unix, Window, MacOS 등 다양한 환경에서 구동이 가능하다.
- R은 우수한 도움말 기능과 그래픽 성능을 가지고 있다.
- R은 프로그래밍 언어이고 사용자 정의 함수를 작성하여 사용할 수 있다.

R의 설치

- R 은 <http://cran.r-project.org/> 에서 다운로드 할 수 있다.
일반적으로 초보자들의 경우 "base"를 설치한다.



The Comprehensive R Archive Network

Frequently used pages

CRAN
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

About R

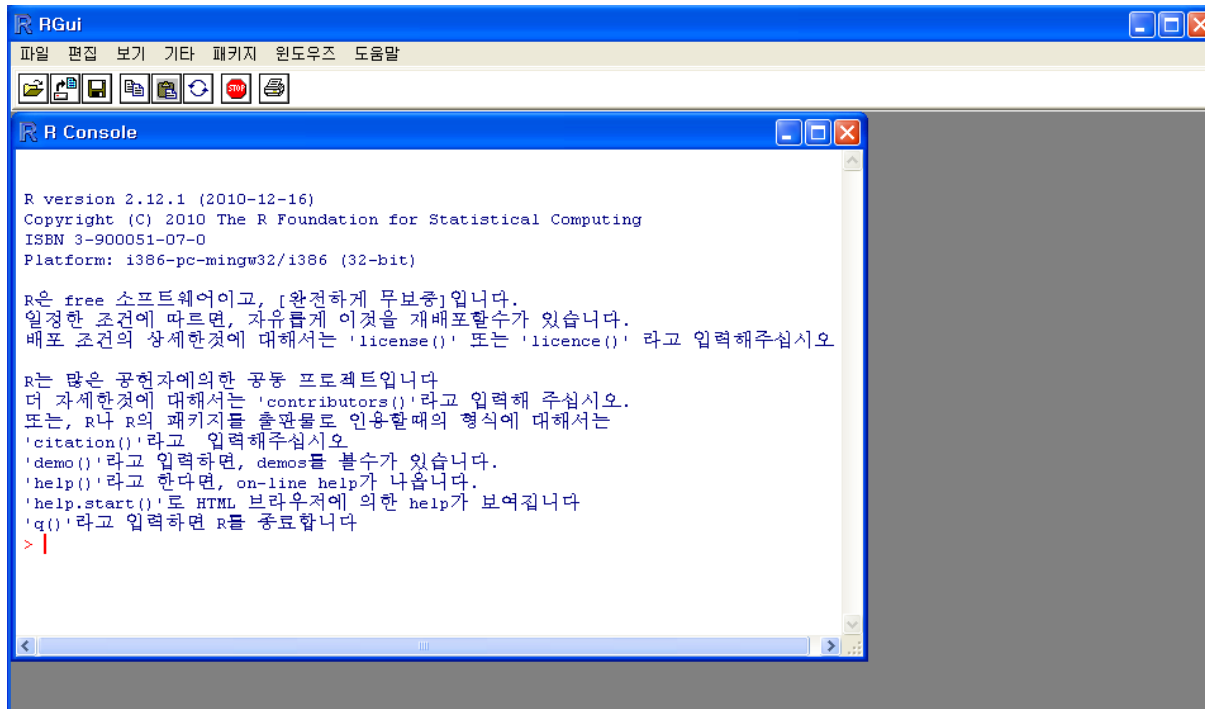
Download and Install R

Precompiled binary distributions of the base system and contributed packages. **Windows and Mac** users most likely want one of these versions of R:

- [Linux](#)
- [MacOS X](#)
- [Windows](#)

초기 화면

- R을 실행하면 다음과 같은 콘솔 창이 열린다.
prompt(>) 에 명령어를 입력한다.



```
R RGui
파일 편집 보기 기타 패키지 윈도우즈 도움말
R R Console
R version 2.12.1 (2010-12-16)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R은 free 소프트웨어이고, [완전하게 무료]입니다.
일정한 조건에 따르면, 자유롭게 이것을 재배포할수가 있습니다.
배포 조건의 상세한것에 대해서는 'license()' 또는 'licence()' 라고 입력해주세요

R는 많은 공헌자에 의한 공동 프로젝트입니다
더 자세한것에 대해서는 'contributors()'라고 입력해 주십시오.
또는, R나 R의 패키지를 출판물로 인용할때의 형식에 대해서는
'citation()'라고 입력해주세요
'demo()'라고 입력하면, demos를 볼수가 있습니다.
'help()'라고 한다면, on-line help가 나옵니다.
'help.start()'도 HTML 브라우저에 의한 help가 보여집니다
'q()'라고 입력하면 R를 종료합니다
> |
```

R의 기본 기능

- "c" 함수를 이용한 자료의 입력

```
> x=c(1,2,3,4,5,6,7,8,9) # 자료를 벡터 형식으로 입력
> x # x를 출력
[1] 1 2 3 4 5 6 7 8 9
```

- 함수들의 사용

```
> mean(x) # 표본평균
[1] 5
> median(x) # 중앙값
[1] 5
> var(x) # 표본분산
[1] 7.5
```

R의 기본 기능

- 변수의 사용

```
> y=x
```

```
# y에 x의 자료를 복사
```

```
> y
```

```
[1] 1 2 3 4 5 6 7 8 9
```

```
> y[5]=0
```

```
# x의 5번째 값에 0을 대입
```

```
> y
```

```
[1] 1 2 3 4 0 6 7 8 9
```

```
> y[6]
```

```
# 벡터 y의 6번째 값 출력
```

```
[1] 6
```

```
> y[-5]
```

```
# 벡터 y의 5번째 값을 제외하고 출력
```

```
[1] 1 2 3 4 6 7 8 9
```

R의 기본 기능

- 변수의 사용

```
> y[c(1,3,5)]
```

```
[1] 1 3 0
```

y의 1, 3, 5번째 값만 출력

```
> y==7
```

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

자료의 값이 7이 되는 곳이 어디인가?

```
> which(y==0)
```

```
[1] 5
```

자료의 값이 0이 되는 곳이 어디인가?

```
> sum(y)
```

```
40
```

y 자료들의 합

```
> sum(y>3)
```

```
5
```

3보다 큰 값의 개수

R의 기본 기능

- 변수의 사용

```
> x+y
```

```
[1] 2 4 6 8 5 12 14 16 18
```

```
> x-y
```

```
[1] 0 0 0 0 5 0 0 0 0
```

```
> x%*%y
```

```
[1]
```

```
[1,] 260
```

```
> max(y); min(y)
```

```
[1] 9
```

```
[1] 0
```

```
# 벡터의 덧셈
```

```
# 벡터의 뺄셈
```

```
# 벡터의 곱셈
```

```
# 자료의 최대값, 최소값
```


R의 기본 기능

- 변수의 사용

```
> y=c(y,10,11,12) # 자료 추가
```

```
> y
```

```
[1] 1 2 3 4 0 6 7 8 9 10 11 12
```

```
> length(y) # 현재 자료의 수
```

```
12
```

```
> y[14:15]=c(14:15) # 자료 추가
```

```
> y
```

```
[1] 1 2 3 4 0 6 7 8 9 10 11 12 NA 14 15
```

```
> ?NA # 도움말 기능 사용 (help search)
```

```
# Not Available / "missing" values
```

R의 기본 기능

- 자료의 수정

> `y=edit(y)`

```
R y - R 편집기
c(1, 2, 3, 4, 0, 6, 7, 8, 9, 10, 11, 12, NA, 14, 15)
```

> `data.entry(y)`

	y	var2	var3	var4	var5	var6	var7
1	1						
2	2						
3	3						
4	4						
5	0						
6	6						
7	7						

R의 기본 기능

- "scan" 함수를 이용한 자료의 입력

```
> x=scan()
```

```
# 콘솔에서 벡터형식 자료 입력 받음
```

```
1: 10 20 30 40 50
```

```
6:
```

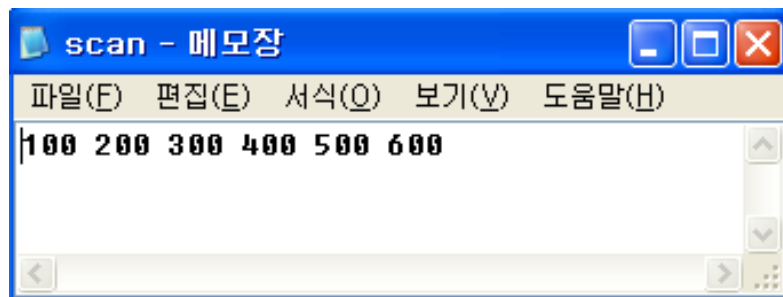
```
Read 5 items
```

```
> x
```

```
[1] 10 20 30 40 50
```

R의 기본 기능

- "scan" 함수를 이용한 자료의 입력



```
> x=scan(file="D:/scan.txt") # scan.txt의 자료를 입력 받음
```

```
Read 6 items
```

```
> x
```

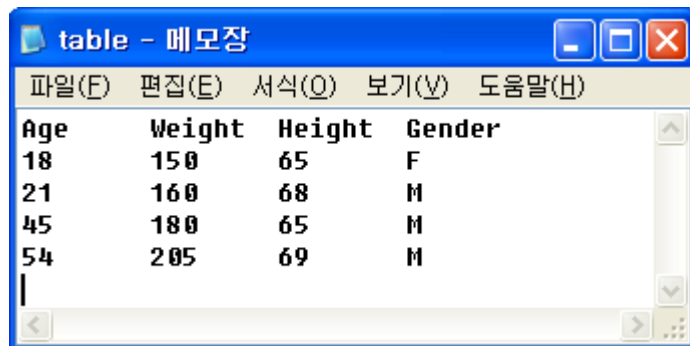
```
[1] 100 200 300 400 500 600
```

```
> x=scan(file="D:/scan.txt") # 값의 구분을 콤마(,)로 입력 받음
```

```
Read 1 items
```

R의 기본 기능

- Table 형식의 자료를 읽기



A screenshot of a Notepad window titled "table - 메모장". The window contains a table with four columns: Age, Weight, Height, and Gender. The data is as follows:

Age	Weight	Height	Gender
18	150	65	F
21	160	68	M
45	180	65	M
54	205	69	M

```
> x=read.table(file="D:/table.txt",header=T)
```

```
> x
```

```
  Age Weight Height Gender
1  18   150    65      F
2  21   160    68      M
3  45   180    65      M
4  54   205    69      M
```

R의 기본 기능

- 고정된 너비의 형식을 갖는 자료(Fixed-Width-Fields) 읽기



```
> x=read.fwf(file="D:/student.txt",widths=c(9,7,4,4,2,4),
+ col.names=c("id","class","section","grades","sem","year"))
> x
```

	id	class	section	grade	sem	year
1	123456789	MTH 214	9872	A	2	2000
2	314159319	MTH 214	9872	B+	2	2000
3	271828232	MTH 214	9872	A-	2	2000

R의 기본 기능

- Spreadsheet형식의 자료를 읽기

```
> x=read.csv(file="data.csv")
```

- XML, url을 통하여 자료 읽기

```
> honk<-read.table("http://www.ats.ucla.edu/stat/examples/alda/honking.csv", sep=";", header=T)
```

```
> honk
```

```
  ID SECONDS CENSOR
1  1    2.88      0
2  2    4.63      1
3  3    2.36      1
```

	A	B	C
1	ID	SECONDS	CENSOR
2	1	2.88	0
3	2	4.63	1
4	3	2.36	1

자료 처리

- 자료 구분

구분			예
수치적 자료 Numerical data Quantitative data	연속형 자료 Continuous data	비율 척도 Ratio scale	키, 몸무게, 나이, 가격
		구간 척도 Interval scale	온도, 지수
	이산형 자료 Discrete data		주사위 결과, 사고 건수
범주형 자료 Categorical data Qualitative data	순위형 자료 Ordinal data		평점, 선호도
	명목형 자료 Nominal data		혈액형, 성별

자료 처리

- 단변량 자료: 범주형 자료

```
> x=c("W","W","H","H","No","No","H","No","No")
```

```
> table(x) # 각 자료의 범주의 도수(frequency)
```

```
x
```

```
H No W
```

```
3 4 2
```

```
> factor(x) # 범주의 수준(level) 또는 인자(factor)
```

```
[1] W W H H No No H No No
```

```
Levels: H No W
```

자료 처리

- 단변량 자료: 범주형 자료

```
> beer=scan()
```

```
1: 3 4 1 1 3 4 3 3 1 3 2 1 2 1 2 3 2 3 1 1 1 1 4 3 1
```

```
26:
```

```
Read 25 items
```

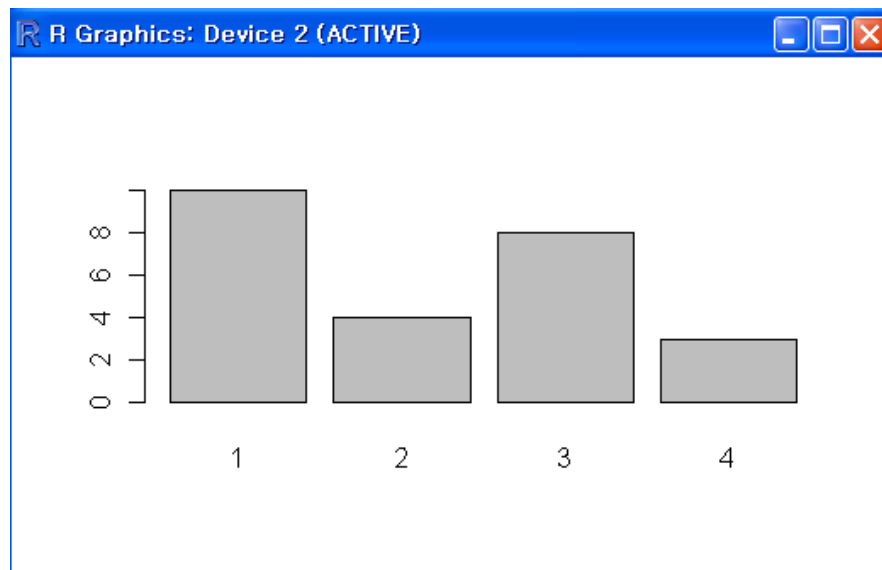
```
> barplot(table(beer))
```

```
> table(beer)/length(beer)
```

```
beer
```

```
 1    2    3    4
```

```
0.40 0.16 0.32 0.12
```



자료 처리

- 단변량 자료: 범주형 자료

```
> table(beer)
```

```
beer
```

```
 1 2 3 4
```

```
10 4 8 3
```

```
> beer.counts=table(beer)
```

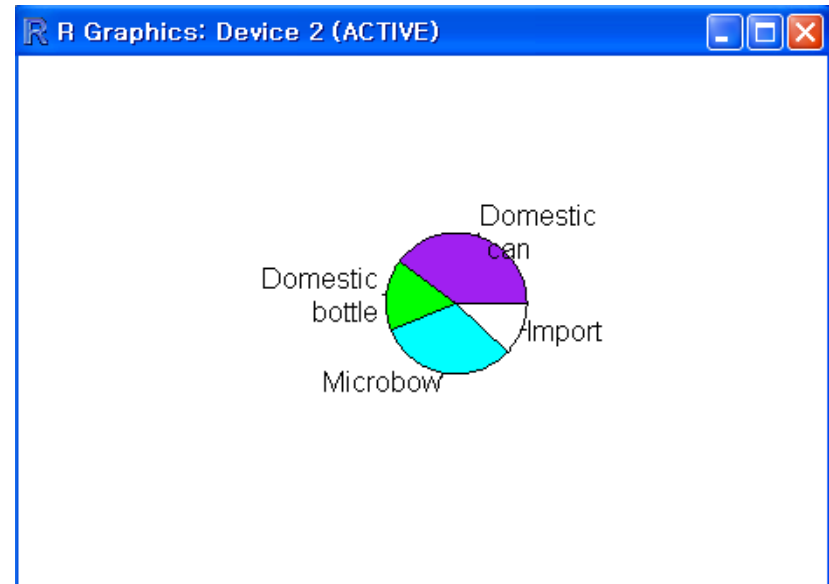
```
# 원형 그래프를 그리고 범주의 이름 및 색을 지정해본다.
```

```
> pie(beer.counts)
```

```
> names(beer.counts)=c("Domestic\n can",  
+ "Domestic\n bottle","Microbow","Import")
```

```
> pie(beer.counts)
```

```
> pie(beer.counts,col=c("purple","green","cyan","white"))
```



자료 처리

- 단변량 자료: 수치형 자료

```
> scores=scan()
```

```
1: 2 3 16 23 14 12 4 13 2 0
```

```
11: 0 0 6 28 31 14 4 8 2 5
```

```
21:
```

```
Read 20 items
```

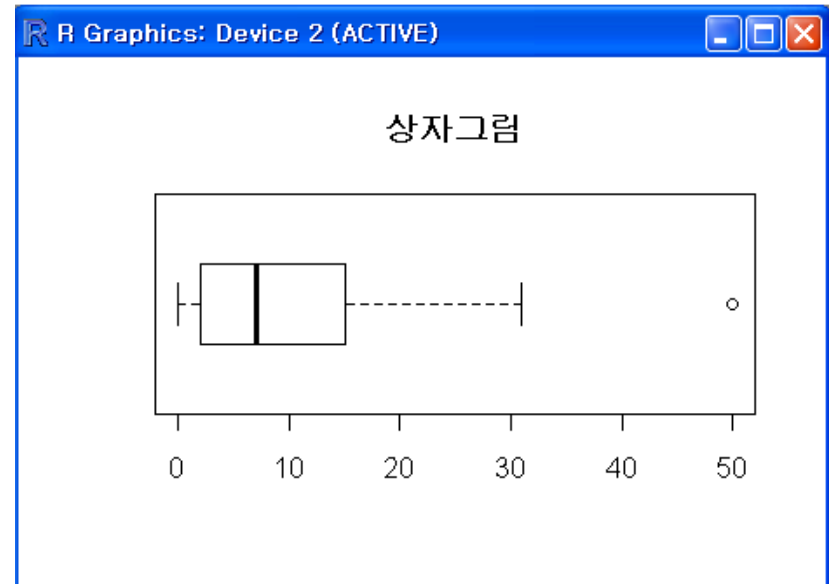
```
# 최소값, 최대값, 사분위수, 상자그림을 확인해본다.
```

```
> summary(scores)
```

```
Min. 1st Qu. Median Mean 3rd Qu. Max.
```

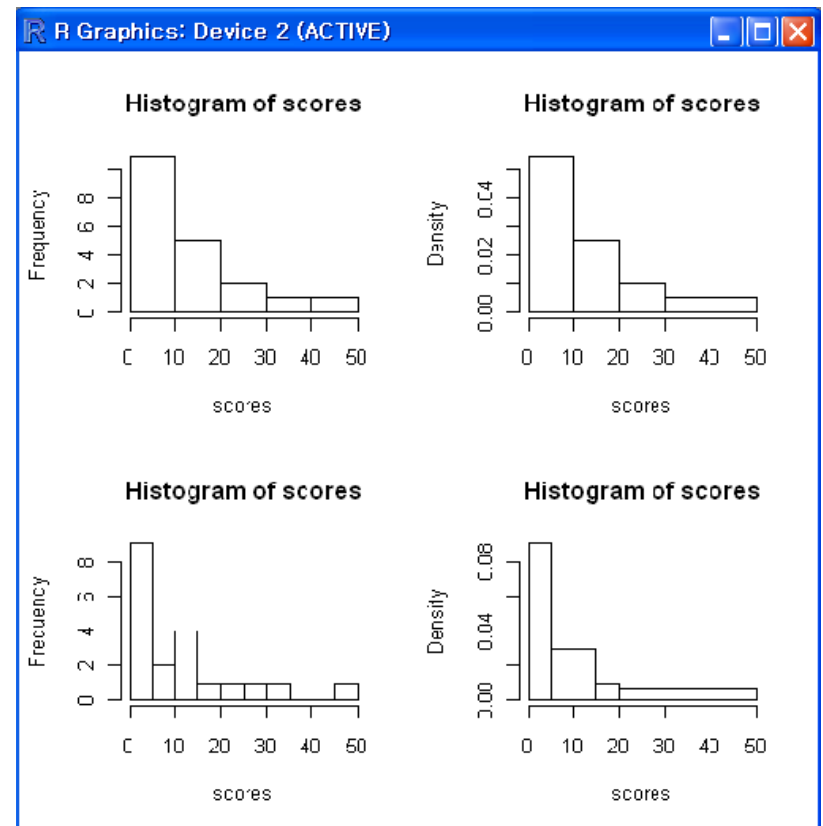
```
0.0    2.0    7.0   11.6   14.5   50.
```

```
> boxplot(scores,main="상자그림",horizontal=T)
```



자료 처리

- 단변량 자료: 수치형 자료
히스토그램을 그려본다.
> `par(mfrow=c(2,2))`
> `hist(scores)`
> `hist(scores,probability=TRUE)`
> `hist(scores,breaks=10)`
> `hist(scores,breaks=`
+ `c(0,5,15,20,max(scores)))`



자료 처리

- 단변량 자료 : 수치형 자료의 범주화

```
> final=scan()
```

```
1: 88 74 60 79 94
```

```
6:
```

```
Read 5 items
```

```
> grades=cut(final,breaks=c(0,60,80,100))
```

```
> grades
```

```
[1] (80,100] (60,80] (0,60] (60,80] (80,100]
```

```
Levels: (0,60] (60,80] (80,100]
```

```
> levels(grades)=c("C","B","A")
```

```
> table(grades)
```

```
grades
```

```
C B A
```

```
1 2 2
```

자료 처리

- 이변량 자료

```
> smokes=c("Y","N","N","Y","N","Y","Y","Y","N","Y")
```

```
> amount=c(1,2,2,3,3,1,2,1,3,2)
```

```
> table(smokes,amount) # 자료를 요약하여 보여줌
```

```
      amount
smokes 1 2 3
  N    0 2 2
  Y    3 2 1
```

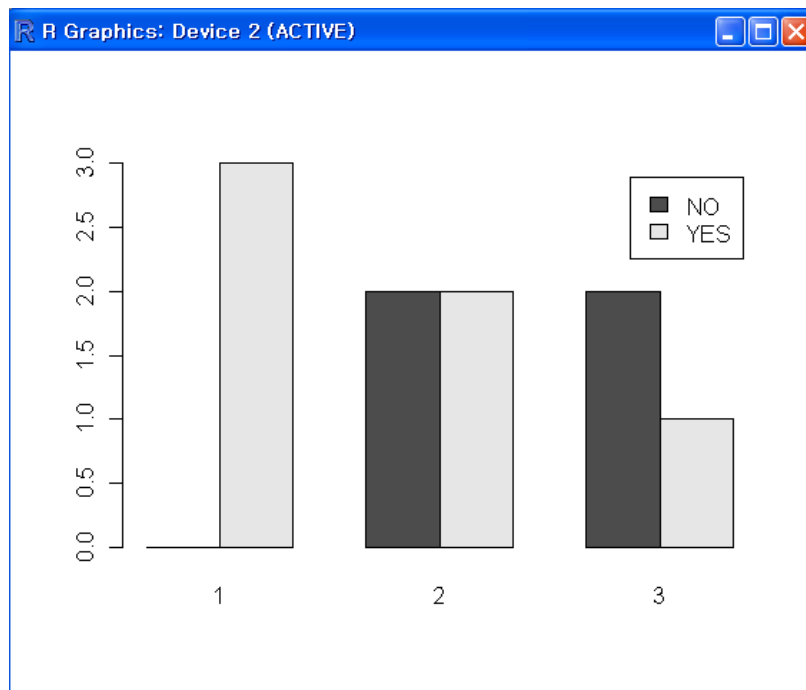
```
> tmp=table(smokes,amount)
```

```
> prop.table(tmp) # 전체에 대한 비율을 구함
```

```
      amount
smokes 1 2 3
  N    0.0 0.2 0.2
  Y    0.3 0.2 0.1
```

자료 처리

- 막대그래프 (bar plot)



```
> barplot(table(smokes,amount),
```

```
+ beside=TRUE,
```

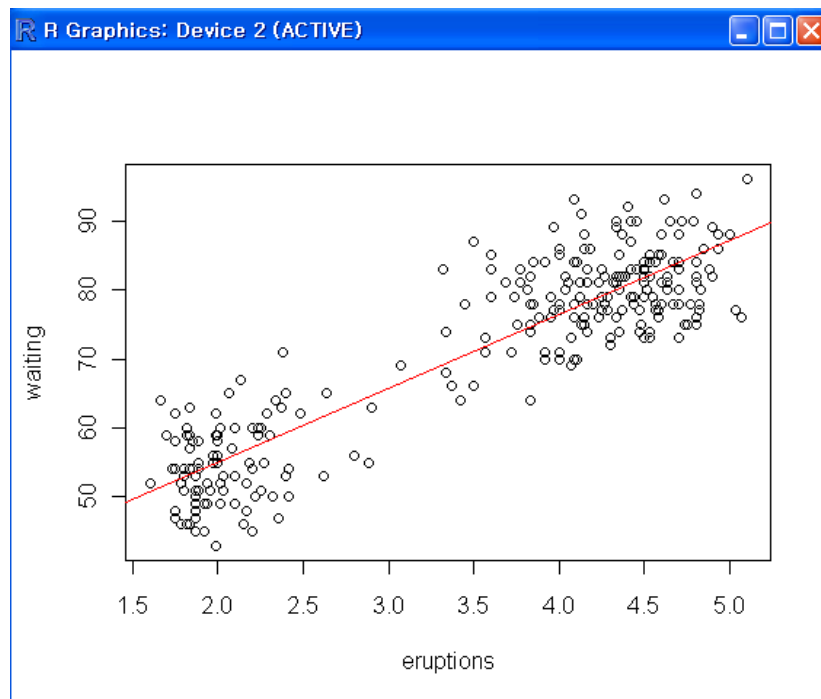
```
+ legend=c("NO","YES"))
```

```
# 범주의 수준에 따라 나란히 그린다.
```

```
# 범례를 삽입
```


자료 처리

- 산점도 (scatterplot)



```
> attach(faithful)
```

```
> plot(eruptions, waiting)
```

```
> abline(lm(waiting~eruptions),col='red') # simple linear regression
```

자료 처리

- 자료 구조에 자료를 구축하기 위해 `data.frame`을 사용한다.

```
> weight=c(150,135,210,140)
```

```
> height=c(65,61,70,65)
```

```
> gender=c("Fe","Fe","M","Fe")
```

```
> study=data.frame(weight,height,gender)
```

```
> row.names(study)=c("Mary","Alice","Bob","Judy")
```

```
> study
```

	weight	height	gender
Mary	150	65	Fe
Alice	135	61	Fe
Bob	210	70	M
Judy	140	65	Fe

자료 처리

- 행 또는 열을 이용하여 배열로서 자료에 접근할 수 있다.

```
> study$weight
```

```
[1] 150 135 210 140
```

```
> study['Mary',]
```

```
      weight height gender
Mary    150     65     Fe
```

```
> study[2:3,]
```

2,3 번째 행에 대한 부분 행렬

```
      weight height gender
Alice   135     61     Fe
Bob     210     70     M
```

```
> study[gender=='M',]
```

성별이 'M'인 행 찾기

```
      weight height gender
Bob     210     70     M
```

자료 처리

- 쌍체 산점도
(paired scatterplot)

```
> n <- 100
```

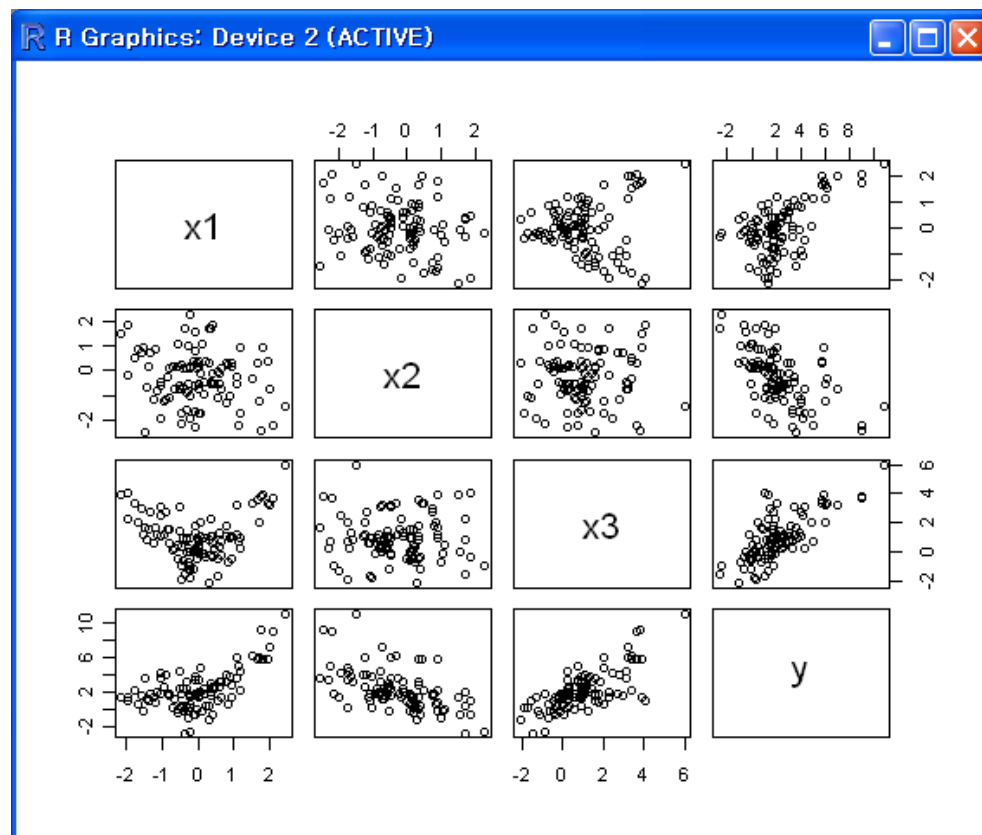
```
> x1 <- rnorm(n)
```

```
> x2 <- rnorm(n)
```

```
> x3 <- x1^2+rnorm(n)
```

```
> y <- 1+x1-x2+x3+.1*rnorm(n)
```

```
> pairs(cbind(x1,x2,x3,y))
```



자료객체의 조작과 관리

- 변수의 생성 및 제거

```
> x1<-1:2
```

```
> x2<-3:4
```

```
> x3<-5:6
```

```
> ls()
```

모든 object들의 이름을 출력

```
[1] "x1" "x2" "x3"
```

```
> rm(x2)
```

object x2를 제거

```
> ls()
```

```
[1] "x1" "x3"
```

```
> rm(list=ls())
```

모든 object들을 제거

```
> ls()
```

```
character(0)
```

자료객체의 조작과 관리

- 자료객체의 형

```
> num=print(c(1, 2, 3))
```

숫자형

```
[1] 1 2 3
```

```
> char=print(c("a","b","c"))
```

문자형

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

```
> logi=print(c(T, F, T))
```

논리형

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

```
> com=print(c(5+3i, 2+4i, 6))
```

복소수형

```
[1] 5+3i 2+4i 6+0i
```

자료객체의 조작과 관리

- 자료객체의 속성: 행렬

```
> matr=matrix(1:9, nrow=3)
```

```
> matr
```

```
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
> dimnames(matr)=list(paste("row",c(1,2,3)),paste("col",c(1,2,3)))
```

```
> matr # 행과 열의 이름을 지정할 수 있다.
```

```
      col 1 col 2 col 3
row 1     1     4     7
row 2     2     5     8
row 3     3     6     9
```

자료객체의 조작과 관리

- 자료객체의 속성: 행렬

```
> length(matr)
```

```
# 자료의 개수
```

```
[1] 9
```

```
> mode(matr)
```

```
# 자료의 형태
```

```
[1] "numeric"
```

```
> dim(matr)
```

```
# 행과 열의 개수
```

```
[1] 3 3
```

```
> dimnames(matr)
```

```
# 행과 열의 이름
```

```
[[1]]
```

```
[1] "row 1" "row 2" "row 3"
```

```
[[2]]
```

```
[1] "col 1" "col 2" "col 3"
```


자료객체의 조작과 관리

- 자료객체의 속성: 배열

```
> arr=array(1:24, c(3,4,2)) # multi-way array
```

```
> arr
```

```
., 1
```

```
  [1] [2] [3] [4]
```

```
[1,]  1  4  7 10
```

```
[2,]  2  5  8 11
```

```
[3,]  3  6  9 12
```

```
., 2
```

```
  [1] [2] [3] [4]
```

```
[1,] 13 16 19 22
```

```
[2,] 14 17 20 23
```

```
[3,] 15 18 21 24
```

자료객체의 조작과 관리

- 자료객체의 속성: 리스트 (서로 다른 형태의 자료를 포함)

```
> a=1:10
```

```
> b=11:15
```

```
> klist=list(vec1=a, vec2=b, descrip="example")
```

```
> length(klist)
```

```
[1] 3
```

```
> klist
```

```
$vec1
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
$vec2
```

```
[1] 11 12 13 14 15
```

```
$descrip
```

```
[1] "example"
```

자료객체의 조작과 관리

- 자료객체의 속성: 리스트

> `length(klist)` # 자료의 개수

[1] 3

> `mode(klist)` # 자료의 형태

[1] "list"

> `names(klist)` # 자료의 이름

[1] "vec1" "vec2" "descrip"

자료객체의 조작과 관리

- 자료객체의 조작

```
> seq(1, by=0.05, along=1:5)
```

```
[1] 1.00 1.05 1.10 1.15 1.20
```

```
> seq(1, 7, by=2)
```

```
[1] 1 3 5 7
```

```
> seq(1, 7, length=3)
```

```
[1] 1 4 7
```

```
> rev(seq(1, 5))
```

```
[1] 5 4 3 2 1
```

```
> rep(c(1, 2, 3), 3)
```

```
[1] 1 2 3 1 2 3 1 2 3
```

```
> paste("no", 1:5)
```

```
[1] "no 1" "no 2" "no 3" "no 4" "no 5"
```

seq() 함수: 계차수열 만듦

자료 순서를 역순으로 만듦

자료를 반복 입력

자료객체의 조작과 관리

- 자료객체의 조작: 벡터

```
> vec1=c(2, 4, 1, 3, 5)
```

```
> vec1[c(2, 3, 5)]
```

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

```
> vec1[c(-2, -3)]
```

```
[1] 2 3 5
```

```
> replace(vec1, 3, 2)
```

3번째 값을 2로 치환

```
[1] 2 4 2 3 5
```

```
> sort(vec1)
```

자료를 오름차순 나열

```
[1] 1 2 3 4 5
```

```
> append(vec1, 8, 4)
```

8을 4번째 값 뒤에 끼워 넣기

```
[1] 2 4 1 3 8 5
```

자료객체의 조작과 관리

- 자료객체의 조작: 행렬

```
> mat=matrix(1:9, ncol=3)
```

열(column)부터 차례로

```
> mat
```

```
  [,1] [,2] [,3]
```

```
[1,]  1  4  7
```

```
[2,]  2  5  8
```

```
[3,]  3  6  9
```

```
> mat=matrix(1:9, ncol=3,byrow=T) # 행(row)부터 차례로
```

```
> mat
```

```
  [,1] [,2] [,3]
```

```
[1,]  1  2  3
```

```
[2,]  4  5  6
```

```
[3,]  7  8  9
```

자료객체의 조작과 관리

- 자료객체의 조작: 행렬

```
> c1=1:3; c2=4:6; c3=7:9
```

```
> cbind(c1,c2,c3)
```

각 벡터를 column으로 갖는 행렬 생성

```
  c1 c2 c3
```

```
[1,] 1  4  7
```

```
[2,] 2  5  8
```

```
[3,] 3  6  9
```

```
> r1=1:3; r2=4:6; r3=7:9
```

각 벡터를 row로 갖는 행렬 생성

```
> rbind(r1,r2,r3)
```

```
  [,1] [,2] [,3]
```

```
r1  1  2  3
```

```
r2  4  5  6
```

```
r3  7  8  9
```

자료객체의 조작과 관리

- 자료객체의 조작: 행렬

```
> y=diag(c(4,2))
```

```
> y
```

```
    [,1] [,2]
```

```
[1,]  4  0
```

```
[2,]  0  2
```

```
> diag(y)
```

```
[1] 4 2
```

- 기타 유용한 함수:

`t(A)`

A의 전치행렬 (transpose)

`solve(A)`

A의 역행렬 (inverse)

`A%*%B`

A와 B의 행렬곱

`eigen(A)`

A의 고유값, 고유벡터

자료객체의 조작과 관리

- 자료객체의 조작: 리스트

```
> li=list("top",c(2,4,6), c(T, F, T))
```

```
> li
```

```
[[1]]
```

```
[1] "top"
```

```
[[2]]
```

```
[1] 2 4 6
```

```
[[3]]
```

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

```
> li[[1]]
```

```
[1] "top"
```

```
> li[[2]][3]
```

```
[1] 6
```

R 프로그래밍과 함수

- 연산자
 - \$: 성분선택
 - ^: 지수, 제곱
 - %%, %/%, %*%: 나머지, 몫, 행렬곱
 - <, >, <=, >=, ==, !=: 비교연산자
 - &, |, &&, ||: 논리 연산자

```
> v1=c(1,2,3)
```

```
> v2=c(2,3,4)
```

```
> v1%0%v2
```

벡터의 외적

```
      [,1] [,2] [,3]
[1,]    2    3    4
[2,]    4    6    8
[3,]    6    9   12
```

R 프로그래밍과 함수

- 기본형식

```
함수이름 = function(함수인수){  
  함수 몸체  
  함수 결과값  
}
```

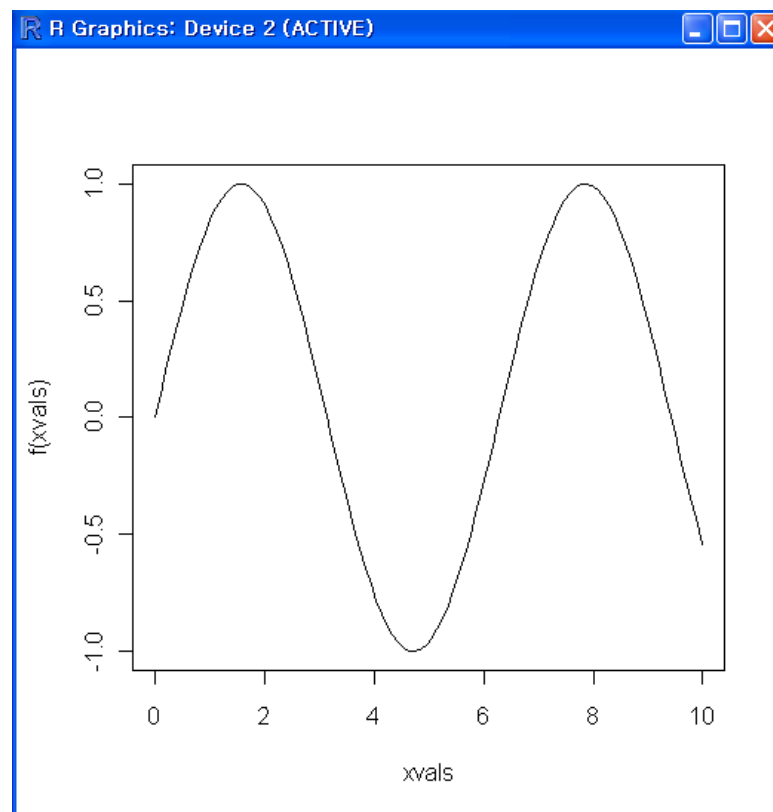
```
> f<-function(x){  
+ x+1  
+ }
```

```
> f(c(2,3))  
[1] 3 4
```

R 프로그래밍과 함수

- 그림을 그려주는 함수

```
> plot.f=function(f,a,b){  
+ xvals=seq(a,b,length=100)  
+ plot(xvals,f(xvals),type="l")}  
> plot.f(sin,0,10)
```



R 프로그래밍과 함수

- n번째 소수를 구하는 함수

```
nth.prime=function(n)
{
  x=1
  count=0
  while(count<n){
    x=x+1
    num.div=0
    for(i in 1:sqrt(x)){
      if (x%%i==0) num.div=num.div+1
      if (num.div>1) break
    }
    if (num.div==1) count=count+1
  }
  print(x)
}
```

```
> for(i in 1:10)
+ nth.prime(i)
[1] 2
[1] 3
[1] 5
[1] 7
[1] 11
[1] 13
[1] 17
[1] 19
[1] 23
[1] 29
```

R 프로그래밍과 함수

- 약수를 구하는 함수

```
divisor=function(n)
{
  if(n<=0)
    print("This number is not positive.")
  else if(n-round(n)!=0)
    print("This number is not integer.")
  else{
    for(i in 1:n){
      if(n%%i==0)
        print(i)
    }
  }
}
```

```
> divisor(-5)
[1] "This number is not positive."
> divisor(0)
[1] "This number is not positive."
> divisor(2.5)
[1] "This number is not integer."
> divisor(12)
[1] 1
[1] 2
[1] 3
[1] 4
[1] 6
[1] 12
```