1강

회귀모형 – R의 활용

정보통계학과 김성수교수

✓ 학습목차

1 R의 소개

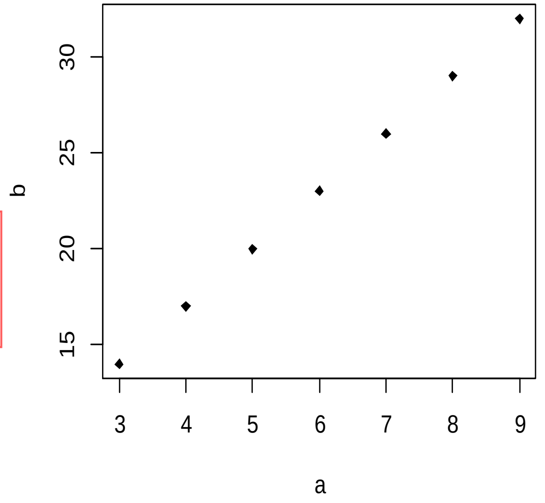
* R의 소개

- R은 자료처리, 통계분석, 통계그래프 등에 뛰어난 기능을 가지고 있는 **무료 통계시스템**
- R은 **대화형 프로그램 언어**(interpreted programming language)
- R은 **객체지향**(object-oriented) 시스템
  - 데이터, 변수, 행렬 등은 모두 객체(object)
  - 객체는 연산자 "<-", 또는 "="에 의해 생성됨

예)

```
> x = 2 : 10
> y = 3*x + 5
> x
[1]  2  3  4  5  6  7  8  9 10
> y
[1] 11 14 17 20 23 26 29 32 35
```

```
> a <- 3:9
> b <- 3*a + 5
> plot(a, b, pch = 18)
```



* R의 태동

S의 탄생

› Becker and Chambers (AT&T Bell Lab) 가 1980년대에 새로 개발한 통계프로그램 언어를 S라 명함 ⇒ S-PLUS 시스템으로 발전

R의 탄생

› Ross Ihaka and Robert Gentleman(Univ. of Auckland, New Zealand)가 교육 목적으로 S의 축소버전(reduced version) "R & R" 을 만듬

R의 발표

› 발표 : 1995년 Martin Maechler가 Ross Ihaka and Robert Gentleman를 설득하여 Linux system과 같이 Open Source Software 규약인 GPL(General Public Licence) 규약 하에 R의 source code를 발표

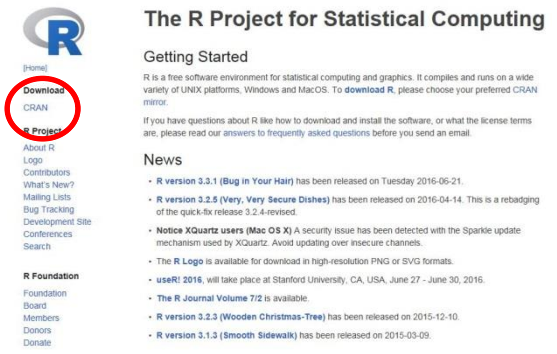
R Core Team의 결성

› 1997년 8월 R 시스템의 발전을 위한 국제적인 R core team이 결성됨. 이후 확장 발전하여 현재(2016년 7월) 20명의 멤버로 구성됨. 2000년 2월 29일 R version 1.0.0 발표됨. 2016년 7월 현재 R version 3.3.1.

참고 www.r-project.org
Peter Dalgaard (2005), Introductory Statistics with R, Springer.

* R 다운 받기

› www.r-project.org 의 CRAN 클릭 ▶ Mirrors 사이트 선택(Korea) ▶Download for Windows ▶ base ▶ Download R …



[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Mailing Lists
Bug Tracking
Development Site
Conferences
Search

R Foundation

Foundation
Board
Members
Donors
Donate

The R Project for Statistical Computing

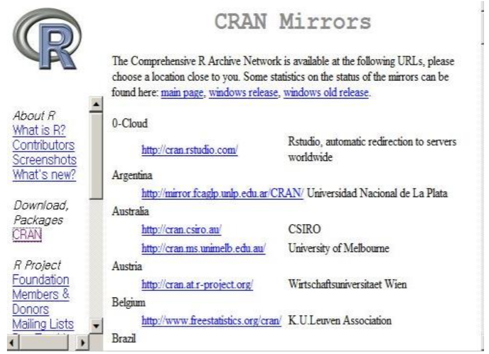
Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- **R version 3.3.1 (Bug in Your Hair)** has been released on Tuesday 2016-06-21.
- **R version 3.2.5 (Very, Very Secure Dishes)** has been released on 2016-04-14. This is a rebadging of the quick-fix release 3.2.4-revised.
- **Notice XQuartz users (Mac OS X)** A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.
- **useR! 2016**, will take place at Stanford University, CA, USA, June 27 - June 30, 2016.
- **The R Journal Volume 7/2** is available.
- **R version 3.2.3 (Wooden Christmas-Tree)** has been released on 2015-12-10.
- **R version 3.1.3 (Smooth Sidewalk)** has been released on 2015-03-09.



About R
What is R?
Contributors
Screenshots
What's new?

Download, Packages
CRAN

R Project
Foundation
Members & Donors
Mailing Lists

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: main page, windows release, windows old release.

0-Cloud

http://cran.rstudio.com/	Rstudio, automatic redirection to servers worldwide

Argentina

http://mirror.fcaglp.unlp.edu.ar/CRAN/	Universidad Nacional de La Plata

Australia

http://cran.csiro.au/	CSIRO
http://cran.ms.unimelb.edu.au/	University of Melbourne

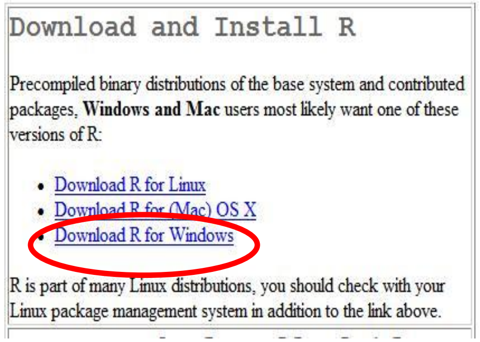
Austria

http://cran.at.r-project.org/	Wirtschaftsuniversitaet Wien

Belgium

http://www.freestatistics.org/cran/	K.U.Leuven Association

Brazil

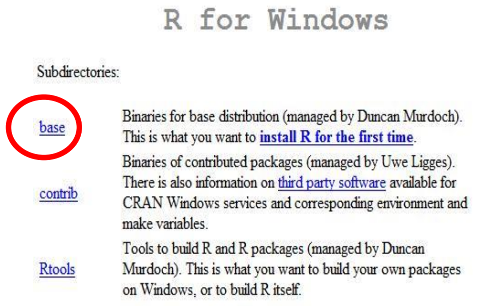


Download and Install R

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

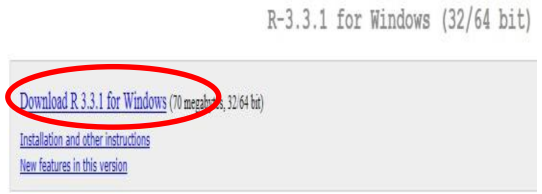
R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.



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	Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.



R-3.3.1 for Windows (32/64 bit)

Download R 3.3.1 for Windows (70 megabytes, 32/64 bit)

Installation and other instructions

New features in this version

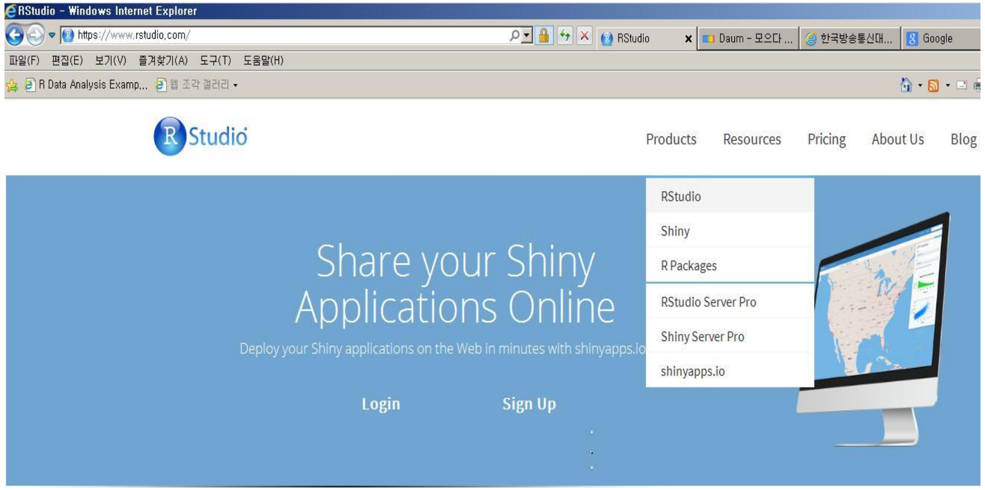
* 작업영역 지정

- **작업 영역(Working directory)** : R에서 데이터를 가져오고 저장하는 디폴트 폴더를 지정해두면 편리하게 작업할 수 있음. 이를 작업 영역(Working directory)이라 함

```
〉 getwd() # shows the working directory
[1] "C:/Users/user/Documents"
〉 setwd("c:/data/reg") # change the working directory
〉 getwd()
[1] "c:/data/reg"
〉 setwd(choose.dir()) # select the working directory interactively
```

* R Studio의 소개

- R Studio : 사용자가 친숙하게 R을 쉽게 사용할 수 있도록 개발된 R 통합환경 시스템
- 다운로드 : www.rstudio.com

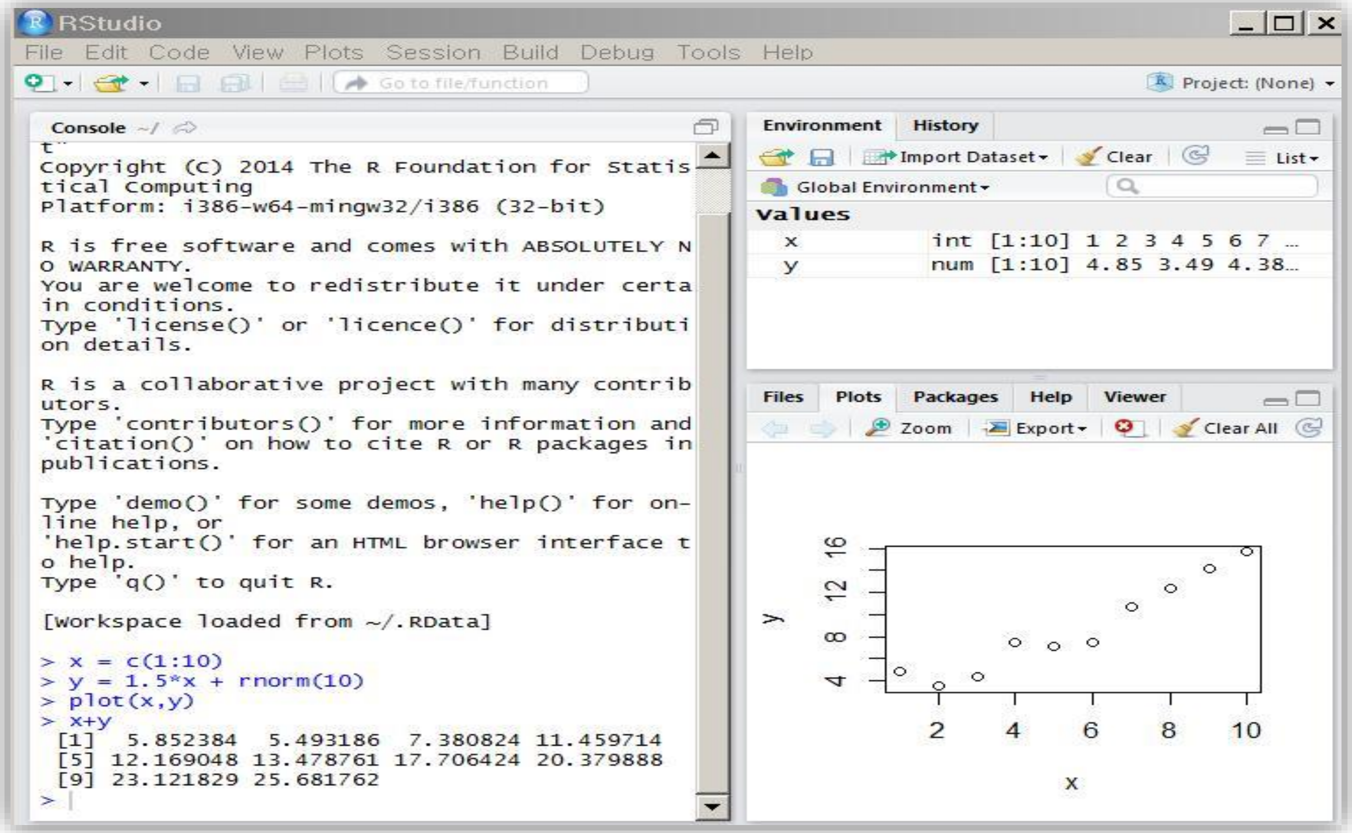


Share your Shiny Applications Online

Deploy your Shiny applications on the Web in minutes with shinyapps.io

참고 http://dss.princeton.edu/training/RStudio101.pdf
http://www.rstudio.com

* R Studio 화면



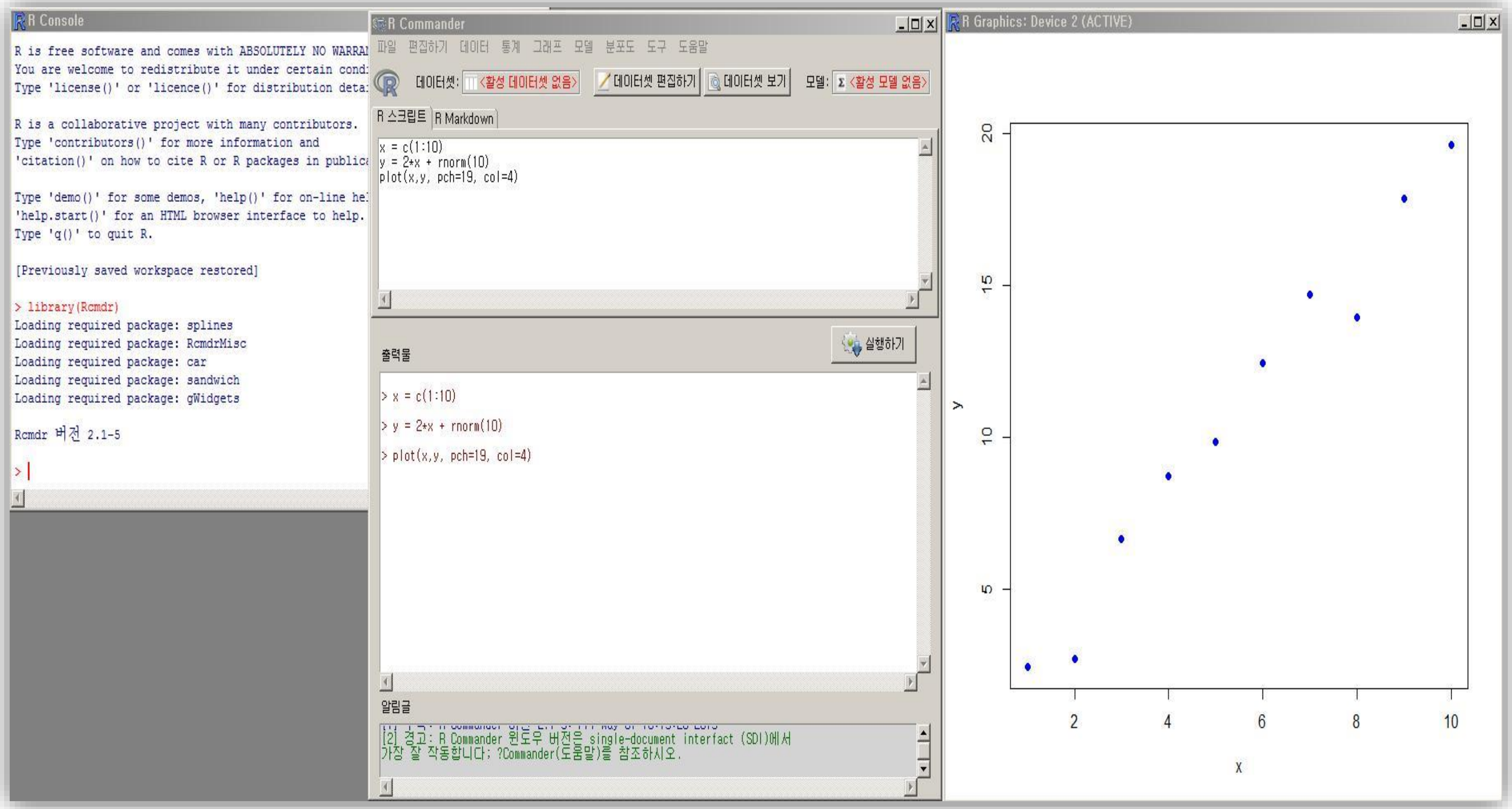
```
t"
Copyright (C) 2014 The R Foundation for Statis
tical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY N
O WARRANTY.
You are welcome to redistribute it under certa
in conditions.
Type 'license()' or 'licence()' for distributi
on details.

R is a collaborative project with many contrib
utors.
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 t
o help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> x = c(1:10)
> y = 1.5*x + rnorm(10)
> plot(x,y)
> x+y
 [1]  5.852384  5.493186  7.380824 11.459714
 [5] 12.169048 13.478761 17.706424 20.379888
 [9] 23.121829 25.681762
> 
```

```
Values
  x        int [1:10] 1 2 3 4 5 6 7 ...
  y        num [1:10] 4.85 3.49 4.38...
```

* R Commander 소개

- R Commander : A GUI for R
  - menu 방식(menu-driven)으로 처리할 수 있도록 개발된 R package
    cf) R is command-driven
  - 개발자 : John Fox (McMaster University)
  - 통계학 입문 코스에 유용하게 이용
  - 복잡한 고급 기능에는 부적합

* R Commander 화면



```
R is free software and comes with ABSOLUTELY NO WARRA
You are welcome to redistribute it under certain cond
Type 'license()' or 'licence()' for distribution deta

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in public

Type 'demo()' for some demos, 'help()' for on-line he
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> library(Rcmdr)
Loading required package: splines
Loading required package: RcmdrMisc
Loading required package: car
Loading required package: sandwich
Loading required package: gWidgets

Rcmdr 버전 2.1-5

> 
```

```
x = c(1:10)
y = 2*x + rnorm(10)
plot(x,y, pch=19, col=4)
```

```
> x = c(1:10)

> y = 2*x + rnorm(10)

> plot(x,y, pch=19, col=4)
```

```
[2] 경고: R Commander 윈도우 버전은 single-document interfact (SDI)에서
가장 잘 작동합니다; ?Commander(도움말)를 참조하시오.
```

2 R의 활용

Vectorized arithmetic

```
> weight <- c(60, 72, 57, 90, 95, 72)
> height <- c(1.75, 1.80, 1.65, 1.90, 1.74, 1.91)
> bmi <- weight/height^2
> bmi
[1] 19.59184 22.22222 20.93664 24.93075 31.37799 19.73630
```

- Calculate mean and SD

```
> xbar <- sum(weight)/length(weight)
> xbar
[1] 74.33333
> sqrt(sum((weight - xbar)^2)/(length(weight) - 1))
[1] 15.42293
```

```
> mean(weight)
[1] 74.33333
> sd(weight)
[1] 15.42293
```

Standard procedures

- Example of t.test()

```
> bmi
[1] 19.59184 22.22222 20.93664 24.93075 31.37799 19.73630
> t.test(bmi, mu=22.5)
        One Sample t-test
data:  bmi
t = 0.3449, df = 5, p-value = 0.7442
alternative hypothesis: true mean is not equal to 22.5
95 percent confidence interval:
 18.41734 27.84791
sample estimates:
mean of x
 23.13262
```

R language essentials

Vectors : character vectors and logical vectors

```
> c("Huey"  "Dewey" "Louie")
[1] "Huey"  "Dewey" "Louie"
> c(T,T,F,T)
[1]  TRUE  TRUE FALSE  TRUE
> bmi
[1] 19.59184 22.22222 20.93664 24.93075 31.37799 19.73630
> bmi > 25
[1] FALSE FALSE FALSE FALSE  TRUE FALSE
```

Quoting and escape sequences

Vectors : character vectors and logical vectors

- Character 연결

```
〉 cat(c("Huey","Dewey","Louie"))
Huey Dewey Louie〉
```

- To get the system prompt onto the next line

```
〉 cat("Huey","Dewey","Louie", "₩n")
Huey Dewey Louie
〉
```

Quoting and escape sequences

Vectors : character vectors and logical vectors

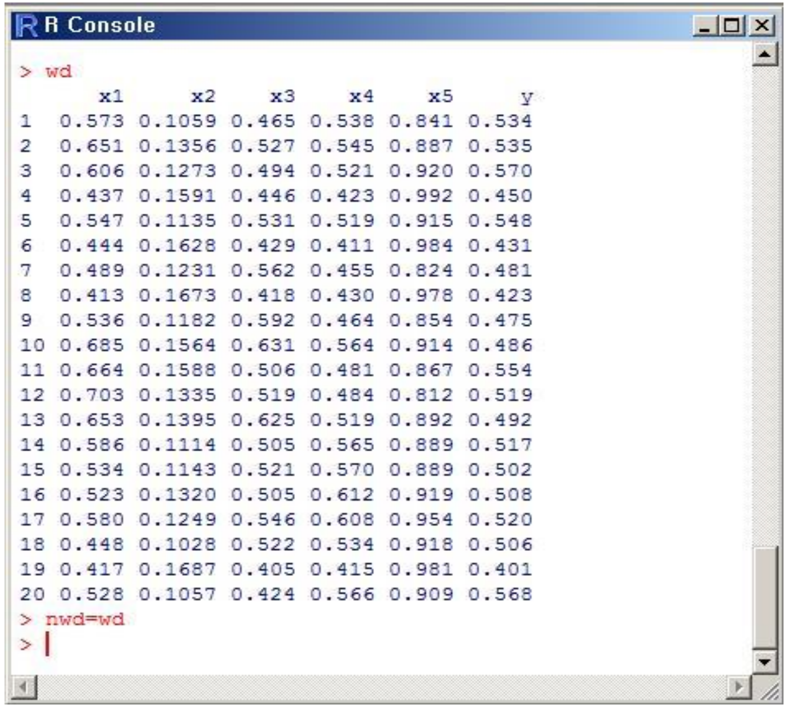
- The backslash (\) is known as the escape character.
- We can insert quote characters with \"

```
> cat("What is \"R\"?\n")
What is "R"?
```

Missing values

- 결측치는 NA 로 표시

```
〉nwd[nwd 〉0.9] = 99
〉nwd[nwd == 99] = NA
〉head(nwd, n=5)
     x1     x2    x3    x4    x5     y
1 0.573     NA 0.465 0.538 0.841 0.534
2 0.651 0.1356 0.527 0.545 0.887 0.535
3 0.606 0.1273 0.494 0.521    NA 0.570
4 0.437 0.1591 0.446 0.423    NA 0.450
5 0.547 0.1135 0.531 0.519    NA 0.548
〉rowSums(is.na(nwd))
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
1 0 1 1 1 1 0 1 0  1  0  0  0  0  0  1  1  2  1  2
〉colSums(is.na(nwd))
x1 x2 x3 x4 x5  y
 0  3  0  0 11  0
〉mywd = na.omit(nwd)
〉head(mywd)
      x1     x2    x3    x4    x5     y
2  0.651 0.1356 0.527 0.545 0.887 0.535
7  0.489 0.1231 0.562 0.455 0.824 0.481
9  0.536 0.1182 0.592 0.464 0.854 0.475
11 0.664 0.1588 0.506 0.481 0.867 0.554
12 0.703 0.1335 0.519 0.484 0.812 0.519
13 0.653 0.1395 0.625 0.519 0.892 0.492
```



```
> wd
      x1     x2    x3    x4    x5     y
1  0.573 0.1059 0.465 0.538 0.841 0.534
2  0.651 0.1356 0.527 0.545 0.887 0.535
3  0.606 0.1273 0.494 0.521 0.920 0.570
4  0.437 0.1591 0.446 0.423 0.992 0.450
5  0.547 0.1135 0.531 0.519 0.915 0.548
6  0.444 0.1628 0.429 0.411 0.984 0.431
7  0.489 0.1231 0.562 0.455 0.824 0.481
8  0.413 0.1673 0.418 0.430 0.978 0.423
9  0.536 0.1182 0.592 0.464 0.854 0.475
10 0.685 0.1564 0.631 0.564 0.914 0.486
11 0.664 0.1588 0.506 0.481 0.867 0.554
12 0.703 0.1335 0.519 0.484 0.812 0.519
13 0.653 0.1395 0.625 0.519 0.892 0.492
14 0.586 0.1114 0.505 0.565 0.889 0.517
15 0.534 0.1143 0.521 0.570 0.889 0.502
16 0.523 0.1320 0.505 0.612 0.919 0.508
17 0.580 0.1249 0.546 0.608 0.954 0.520
18 0.448 0.1028 0.522 0.534 0.918 0.506
19 0.417 0.1687 0.405 0.415 0.981 0.401
20 0.528 0.1057 0.424 0.566 0.909 0.568
> nwd=wd
> |
```

– rowSums(is.na(wd)) : 각 행별로 결측치의 수를 나타냄
– colSums(is.na(wd)) : 각 열별로 결측치의 수를 나타냄
– na.omit(wd) : 결측치를 제거

Functions that create vectors

- 벡터 객체 연결

```
> x <- c(1, 2, 3)
> y <- c(10, 20)
> c(x,y,5)
[1]  1  2  3 10 20  5
```

- All elements of a vector have the same type. If you concatenate vectors ofdifferent types, they will be converted to the least “restrictive” type:

```
> c(FALSE, 3)
[1] 0 3
> c(pi, "abc")
[1] "3.14159265358979" "abc"
> c(FALSE, "abc")
[1] "FALSE" "abc"
```

Functions that create vectors

- equidistant series of numbers

```
> seq(4,9)
[1] 4 5 6 7 8 9
```

- Generate repeated values

```
> oops <- c(7,9,13)
> rep(oops,3)
[1]  7  9 13  7  9 13  7  9 13
> rep(oops,1:3)
[1]  7  9  9 13 13 13
> rep(1:2,c(10,15))
[1] 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Matrices and arrays

- equidistant series of numbers

```
〉 x <– 1:12
〉 dim(x) = c(3,4)
〉 x
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
〉 matrix(1:12,nrow=3,byrow=T)
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

- rownames (colnames)

```
〉 rownames(x) <– LETTERS[1:3]
〉 x
  [,1] [,2] [,3] [,4]
A    1    4    7   10
B    2    5    8   11
C    3    6    9   12
〉 t(x)
      A  B  C
[1,]  1  2  3
[2,]  4  5  6
[3,]  7  8  9
[4,] 10 11 12
```

Matrices and arrays

행렬의 곱

```
> A <- x[, c(1:2)]
> A
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
> ATA <- t(A) %*% A
> ATA
     [,1] [,2]
[1,]   14   32
[2,]   32   77
```

역행렬 (inverse matrix)

```
> ATAI <- solve(ATA)
> ATAI
           [,1]       [,2]
[1,]  1.4259259 -0.5925926
[2,] -0.5925926  0.2592593
> ATAI %*% ATA
     [,1]          [,2]
[1,]    1 -8.881784e-16
[2,]    0  1.000000e+00
```

cbind, rbind

```
> cbind(A=1:4,B=5:8,C=9:12)
     A B  C
[1,] 1 5  9
[2,] 2 6 10
[3,] 3 7 11
[4,] 4 8 12
> rbind(A=1:4,B=5:8,C=9:12)
  [,1] [,2] [,3] [,4]
A    1    2    3    4
B    5    6    7    8
C    9   10   11   12
> aa = cbind(A=1:4,B=5:8,C=9:12)
> cbind(1, aa)
       A B  C
[1,] 1 1 5  9
[2,] 1 2 6 10
[3,] 1 3 7 11
[4,] 1 4 8 12
```

factors

- Categorical variables : factors

```
> pain <- c(0,3,2,2,1)
> fpain <- factor(pain,levels=0:3)
> levels(fpain) <-
c("none","mild","medium","severe")
> fpain
[1] none   severe medium medium mild
Levels: none mild medium severe
> as.numeric(fpain)
[1] 1 4 3 3 2
> levels(fpain)
[1] "none"   "mild"   "medium" "severe"
```

Data entry

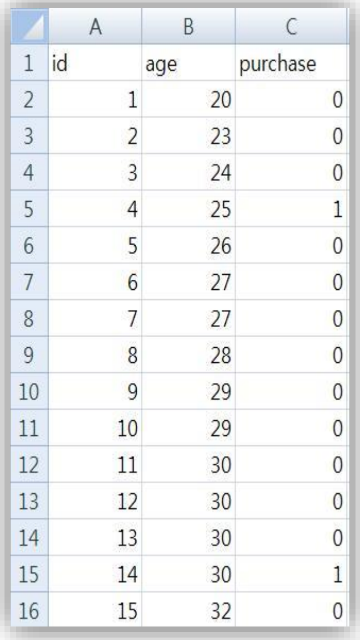
- Read data file
  - text file : read.table, read.csv
  - excel file : read.xlsx (package : xlsx)

- Ex) reading excel file

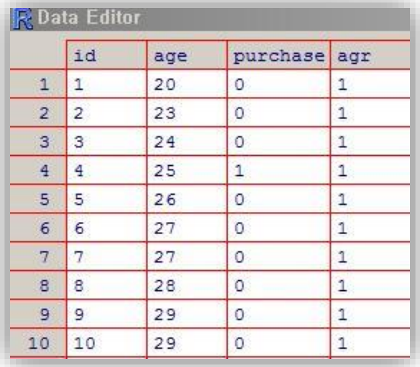
```
> install.packages("xlsx")
> library(xlsx)
> drug.data = read.xlsx("c:/data/drug.xlsx", 1)
```

- The data editor

```
> edit(drug.data)
```



	A	B	C
1	id	age	purchase
2	1	20	0
3	2	23	0
4	3	24	0
5	4	25	1
6	5	26	0
7	6	27	0
8	7	27	0
9	8	28	0
10	9	29	0
11	10	29	0
12	11	30	0
13	12	30	0
14	13	30	0
15	14	30	1
16	15	32	0



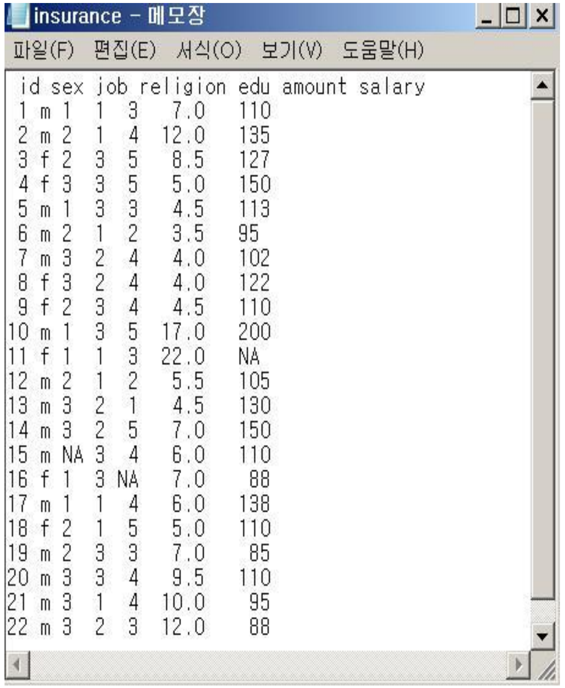
	id	age	purchase	agr
1	1	20	0	1
2	2	23	0	1
3	3	24	0	1
4	4	25	1	1
5	5	26	0	1
6	6	27	0	1
7	7	27	0	1
8	8	28	0	1
9	9	29	0	1
10	10	29	0	1

▪ 숫자로 입력된 값을 라벨로 바꾸기

예) 변수 job 1 = 근로자, 2 = 사무직, 3 = 전문가
    edu 1 = 무학, 2 = 국졸, 3 = 중졸, 4 = 고졸, 5 = 대졸

```
> insurance = read.table("c:/data/insurance.txt", header=T)
> insurance$job = factor(insurance$job, levels=c(1:3),
                         labels=c("근로자","사무직","전문가"))
> insurance$edu2 = ordered(insurance$edu, levels=c(1:5),
                         labels=c("무학","국졸","중졸","고졸","대졸"))
> head(insurance)
 id sex    job religion edu amount salary edu2
1 1   m 근로자        1   3    7.0    110 중졸
2 2   m 사무직        1   4   12.0    135 고졸
3 3   f 사무직        3   5    8.5    127 대졸
4 4   f 전문가        3   5    5.0    150 대졸
5 5   m 근로자        3   3    4.5    113 중졸
6 6   m 사무직        1   2    3.5     95 국졸
```

– 명목형(nominal data) : factor() 함수
– 순서형(ordered data) : ordered() 함수



```
 id sex job religion edu amount salary
 1 m 1  1  3   7.0  110
 2 m 2  1  4  12.0  135
 3 f 2  3  5   8.5  127
 4 f 3  3  5   5.0  150
 5 m 1  3  3   4.5  113
 6 m 2  1  2   3.5  95
 7 m 3  2  4   4.0  102
 8 f 3  2  4   4.0  122
 9 f 2  3  4   4.5  110
10 m 1  3  5  17.0  200
11 f 1  1  3  22.0  NA
12 m 2  1  2   5.5  105
13 m 3  2  1   4.5  130
14 m 3  2  5   7.0  150
15 m NA 3  4   6.0  110
16 f 1  3 NA   7.0   88
17 m 1  1  4   6.0  138
18 f 2  1  5   5.0  110
19 m 2  3  3   7.0   85
20 m 3  3  4   9.5  110
21 m 3  1  4  10.0   95
22 m 3  2  3  12.0   88
```

list

- list() : to combine a collection of objects into a larger composite object

```
> intake.pre <- c(5260,5470,5640,6180,6390)
> intake.post <- c(3910,4220,3885,5160,5645)
> mylist <- list(before=intake.pre,after=intake.post)
> mylist
$before
[1] 5260 5470 5640 6180 6390

$after
[1] 3910 4220 3885 5160 5645

> mylist$before
[1] 5260 5470 5640 6180 6390
```

Data frame

```
〉 intake.pre 〈– c(5260,5470,5640,6180,6390)
〉 intake.post 〈– c(3910,4220,3885,5160,5645)
〉 d 〈– data.frame(intake.pre,intake.post)
〉 d
  intake.pre intake.post
1       5260        3910
2       5470        4220
3       5640        3885
4       6180        5160
5       6390        5645
〉 d$intake.pre
[1] 5260 5470 5640 6180 6390
```

Indexing

```
> intake.pre <- c(5260,5470,5640,6180,6390)
> intake.post <- c(3910,4220,3885,5160,5645)
> d <- data.frame(intake.pre,intake.post)
> d
  intake.pre intake.post
1       5260        3910
2       5470        4220
3       5640        3885
4       6180        5160
5       6390        5645
> d$intake.pre
[1] 5260 5470 5640 6180 6390
> intake.pre[c(1,5)]    # indexing
[1] 5260 6390
> v <- c(1,5)
> intake.pre[v]    # indexing
[1] 5260 6390
> intake.pre[-c(1,5)]
[1] 5470 6180 6390
```

Conditional selection

```
> intake.pre <- c(5260,5470,5640,6180,6390)
> intake.post <- c(3910,4220,3885,5160,5645)
> d <- data.frame(intake.pre,intake.post)
> d
  intake.pre intake.post
1       5260        3910
2       5470        4220
3       5640        3885
4       6180        5160
5       6390        5645
> intake.post[intake.pre > 6000]
[1] 5160 5645
> intake.post[intake.pre > 5500 & intake.pre <= 6200]
[1] 3885 5160
> d[d$intake.pre > 6000,]   # conditional selection
  intake.pre intake.post
4       6180        5160
5       6390        5645
```

Implicit loops

```
> library(ISwR)
> data(thuesen)
> head(thuesen, 3)
  blood.glucose short.velocity
1          15.3           1.76
2          10.8           1.34
3           8.1           1.27
> lapply(thuesen, mean, na.rm=T)
$blood.glucose
[1] 10.3

$short.velocity
[1] 1.325652

> sapply(thuesen, mean, na.rm=T)
 blood.glucose short.velocity
     10.300000       1.325652
```

```
> m = matrix(rnorm(12), 4)
> m
           [,1]        [,2]       [,3]
[1,] 0.39222840 -1.03034262  1.2108641
[2,] 0.02383732 -1.42274527 -0.8772476
[3,] 1.27594439  0.52841605 -0.9904278
[4,] 0.20806262  0.03695426  0.3207577
> apply(m, 2, min)
[1]  0.02383732 –1.42274527 –0.99042785

> head(energy, 3)
 expend stature
1   9.21   obese
2   7.53    lean
3  7.48    lean
> tapply(energy$expend, energy$stature, median)
 lean obese
 7.90  9.69
```

Sorting

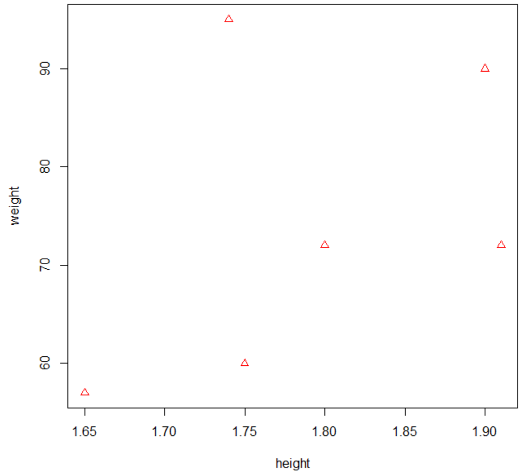
```
> intake
   pre  post
1  5260 3910
2  5470 4220
3  5640 3885
4  6180 5160
5  6390 5645
6  6515 4680
7  6805 5265
8  7515 5975
9  7515 6790
10 8230 6900
11 8770 7335
> o1 <- order(intake$post)
> o1
 [1]  3  1  2  6  4  7  5  8  9 10 11
> intake$post[o1]
 [1] 3885 3910 4220 4680 5160 5265 5645 5975 6790 6900 7335
> intake$pre[o1]
 [1] 5640 5260 5470 6515 6180 6805 6390 7515 7515 8230 8770
```

3 R 그래프

Graphics

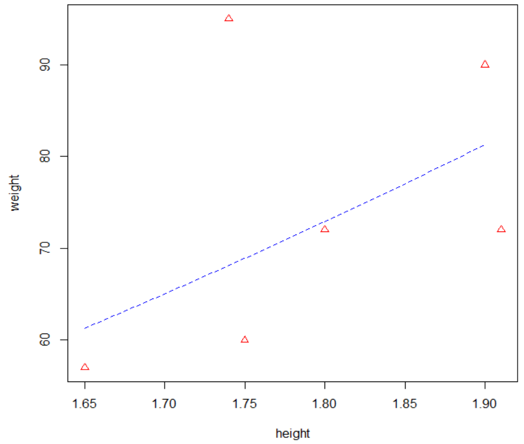
- Example of plot()

```
> plot(height, weight, pch=2, col="RED")
```



- Superimpose of curve

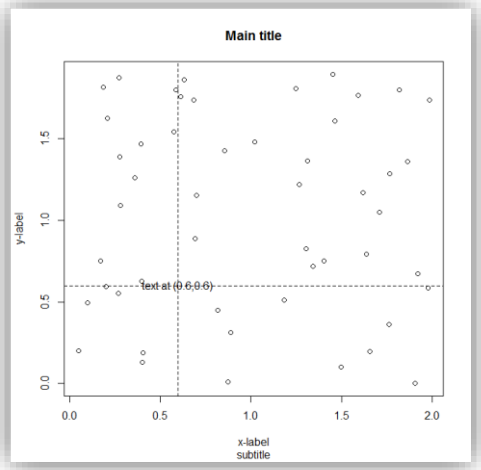
```
> hh <- c(1.65, 1.70, 1.75, 1.80, 1.85, 1.90)
> lines(hh, 22.5 * hh^2, col="BLUE", lty=2)
```

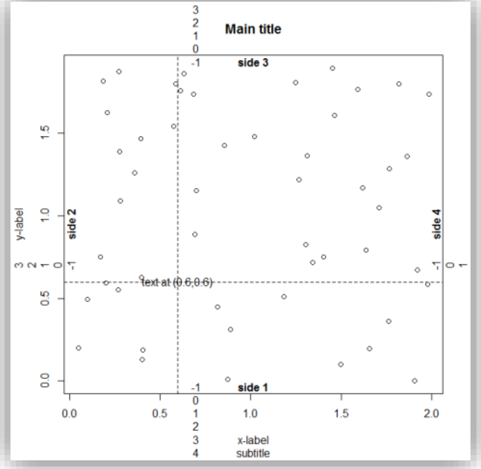


The graphics system

- Plot layout

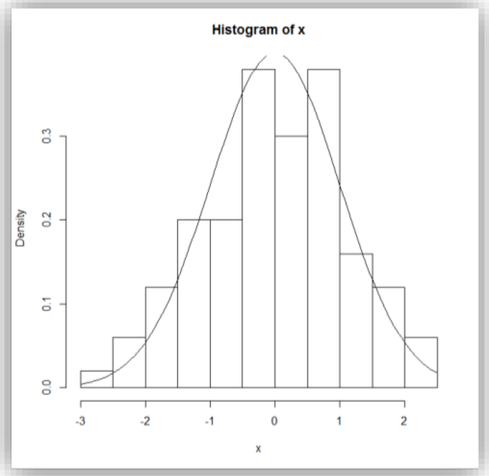
```
> x <- runif(50,0,2)
> y <- runif(50,0,2)
> plot(x, y, main="Main title", sub="subtitle",
+            xlab="x-label", ylab="y-label")
> text(0.6,0.6,"text at (0.6,0.6)")
> abline(h=.6,v=.6, lty=2)

> for (side in 1:4) mtext(-1:4,side=side,at=.7,line=-1:4)
> mtext(paste("side",1:4), side=1:4, line=-1, font=2)
```

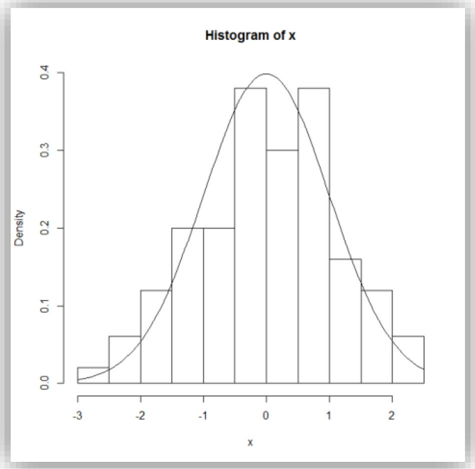




Combining plots

```
> # 1
> x <- rnorm(100)
> hist(x,freq=F)
> curve(dnorm(x),add=T)
> # 2
> h <- hist(x, plot=F)
> ylim <- range(0, h$density, dnorm(0))
> hist(x, freq=F, ylim=ylim)
> curve(dnorm(x), add=T)
```





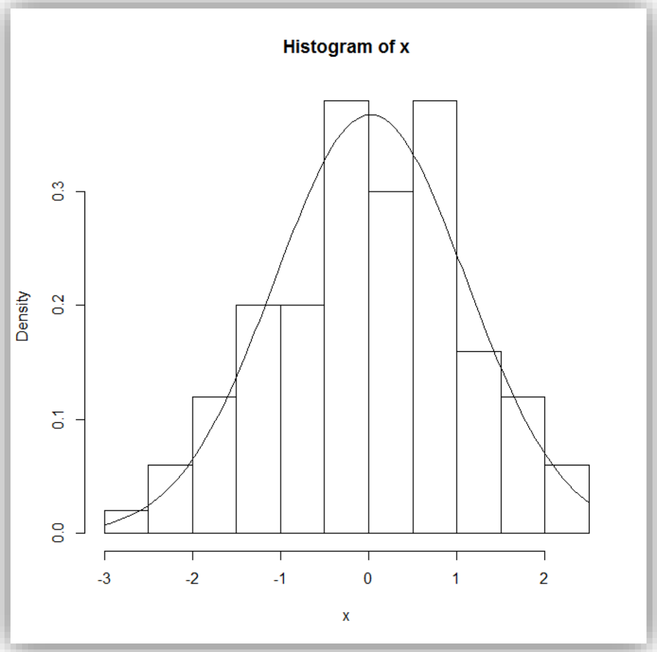
R programming

- 프로그램파일 : c:/rpgm/p44.r

```
hist.normal <-  function(x)
{
  h <- hist(x, plot=F)
  s <- sd(x)
  m <- mean(x)
  ylim <- range(0,h$density,dnorm(0,sd=s))
  hist(x, freq=F, ylim=ylim)
  curve(dnorm(x,m,s), add=T)
}
```

- 실행

```
> source("c:/rpgm/p44.r")
> x = rnorm(100)
> hist.normal(x)
```



다음시간 안내

2강. 단순회귀모형(1)