02

데이터분석방법론(1)

R개요

통계데이터과학과 장영재 교수

학습목차

1 R 시작하기

2 R 기초

3 R 데이터구조

4 R 데이터 편집

01

R 시작하기

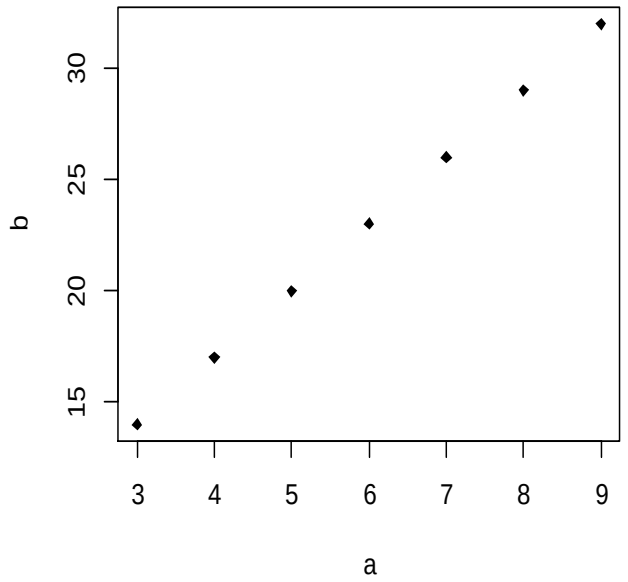
1. R의 소개

- 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)
```



2. R의 태동

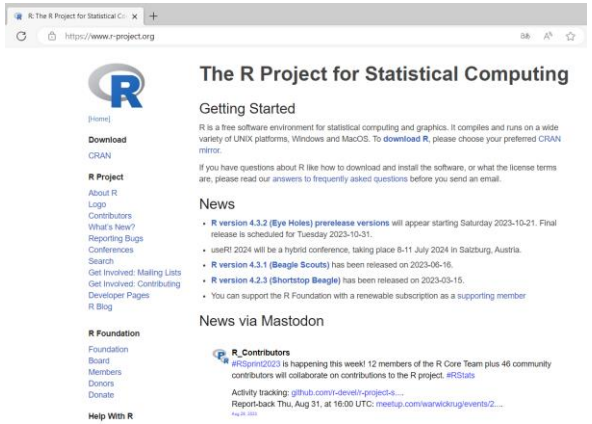
- **S의 탄생** : Becker and Chambers (AT&T Bell Lab) 가 1980년대에 새로 개발한 통계프로그램 언어를 S 라 명함 – S-PLUS 시스템으로 발전.
- **R의 탄생** : Ross Ihaka and Robert Gentleman(Univ. of Auckland, New Zealand) 가 교육 목적으로 S 의 축소버전 (reduced version) "R & R" 을 만듦
- **R의 발표** : 1995년 Martin Maechler가 Ross Ihaka and Robert Gentleman를 설득하여 Linux system 과 같이 Open Source Software 규약인 GPL(General Public Licence) 규약하에 R의 source code를 발표
- **R Core Team의 결성** : 1997년 8월 R 시스템의 발전을 위한 국제적인 R core team이 결성됨. 이후 확장 발전하여 현재(2015년 7월) 21명의 멤버로 구성됨. 2000년 2월 29일 R version 1.0.0 발표됨. 2015년 7월 현재 R version 3.2.1.

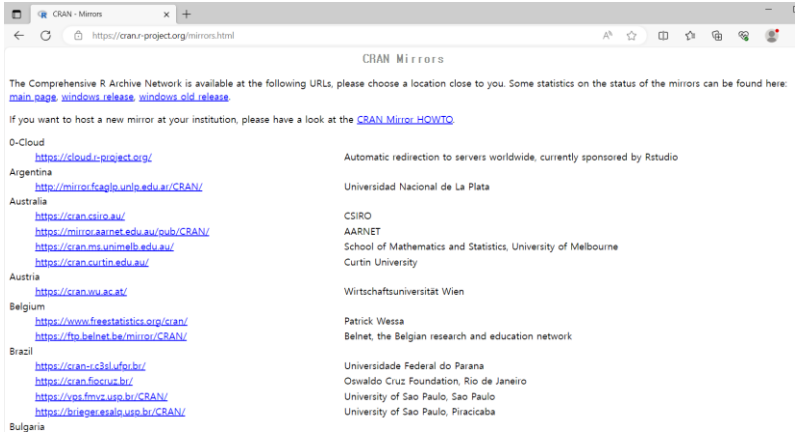
❖ 참고 : www.r-project.org

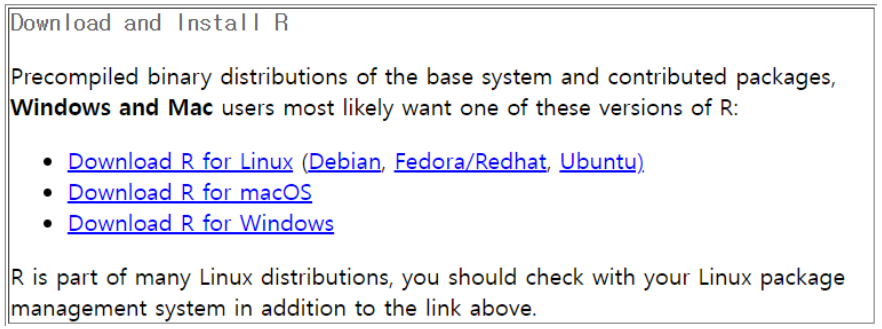
Peter Dalgaard (2005), Introductory Statistics with R, Springer.

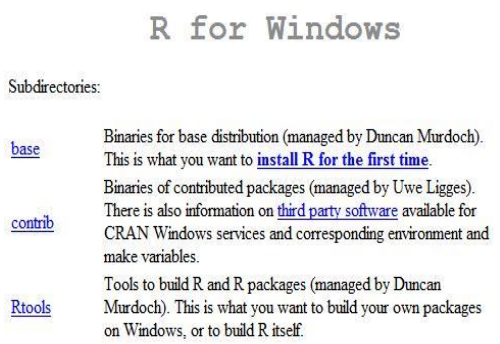
3. R다운받기

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









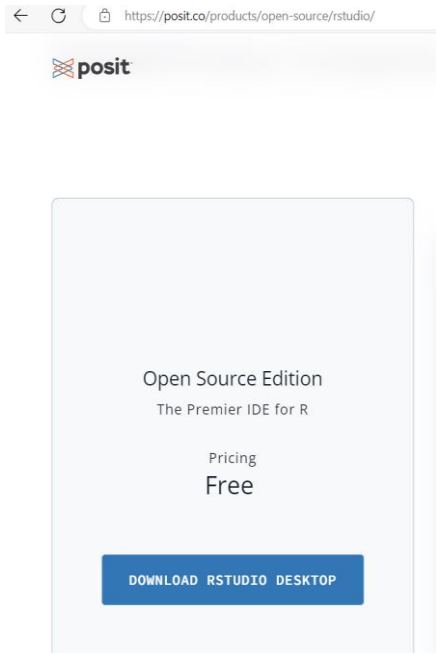
4. 작업영역 지정

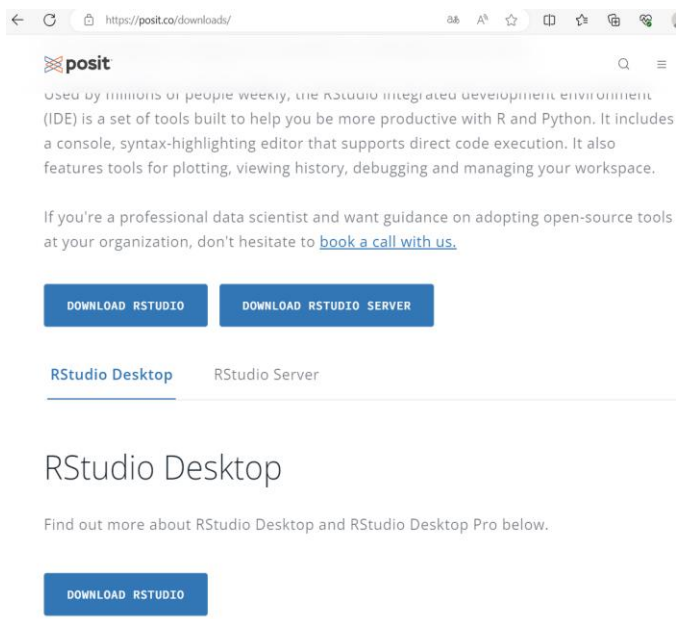
- 작업 영역(Working directory) : R에서 데이터를 가져오고 저장하는 디폴트 폴더을 지정해두면 편리하게 작업할 수 있음. 이를 작업 영역(Working directory) 이라 함.

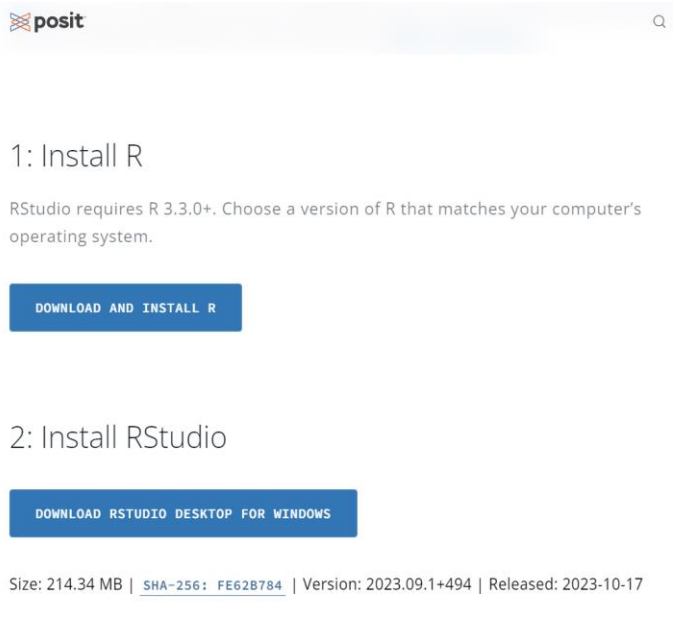
```
> getwd() # shows the working directory
 [1] "C:/Users/user/Documents"
> setwd("c:/grad/data") # change the working directory
> getwd()
 [1] "c:/grad/data"
> setwd(choose.dir()) # select the working directory interactively
```

5. R Studio의 소개

- R Studio : 사용자가 친숙하게 R을 쉽게 사용할 수 있도록 개발된 R 통합환경 시스템
- 다운로드 : www.rstudio.com → posit.co/products/open-source/rstudio/







6. R Studio 화면

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function

Project: (None)

Console ~/

```
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
>
```

Environment History

Import Dataset Clear List

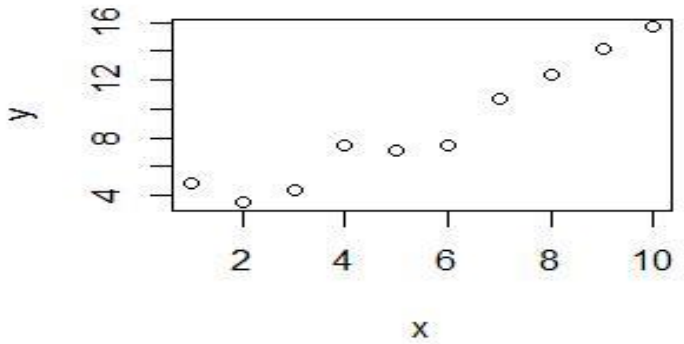
Global Environment

Values

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

Files Plots Packages Help Viewer

Zoom Export Clear All



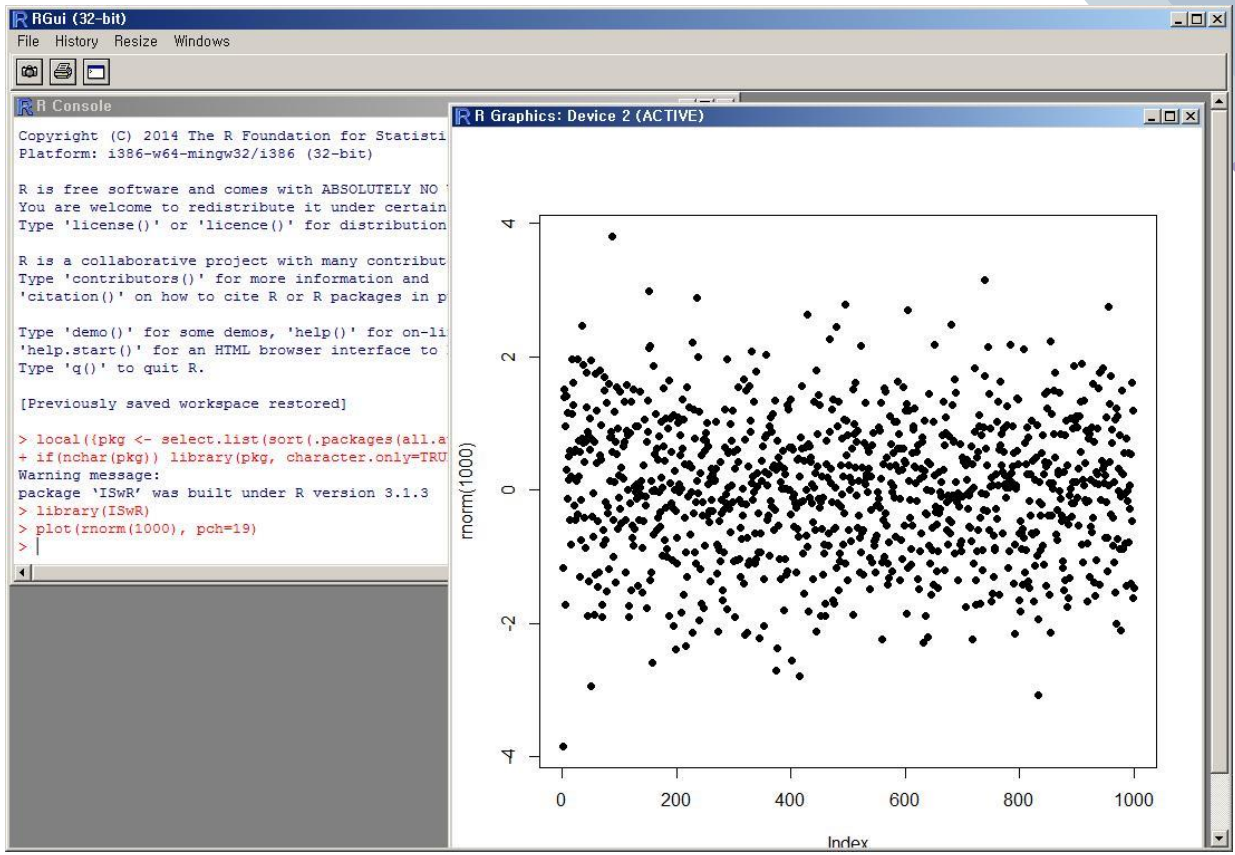
02

R 기초

1. First Steps

- ISwR 패키지 설치
  - ① [Packages] - [package install]을 선택 또는
    > install.packages("ISwR")
  - ② Cran Mirror 선택
  - ③ 패키지 ISwR 선택

- 가동
  > library(ISwR)
  > plot(rnorm(1000), pch=19)



2. 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
```

3. Standard Procedures

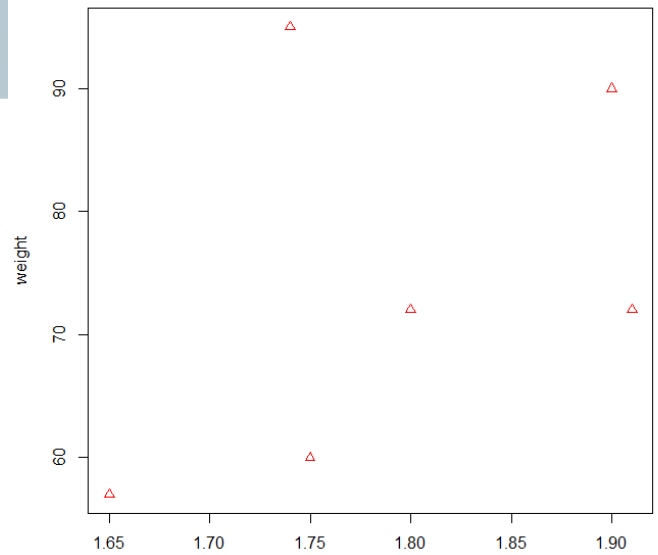
- Example of t.test()

```
> 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
```

4. Graphics

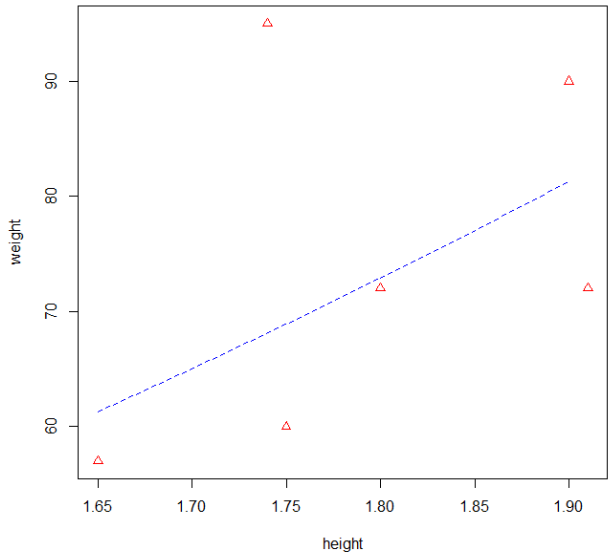
- Example of plot()

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



- Superimpose of a 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)
```



5. R language Essentials

Vectors : character vectors and logical vectors

```
 > c("Huey","Dewey","Louie")
[1] "Huey"  "Dewey" "Louie"
> 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
```

6. Quoting and Escape Sequences

◆ 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

>

◆ The backslash (₩) is known as the escape character.
◆ We can insert quote characters with ₩

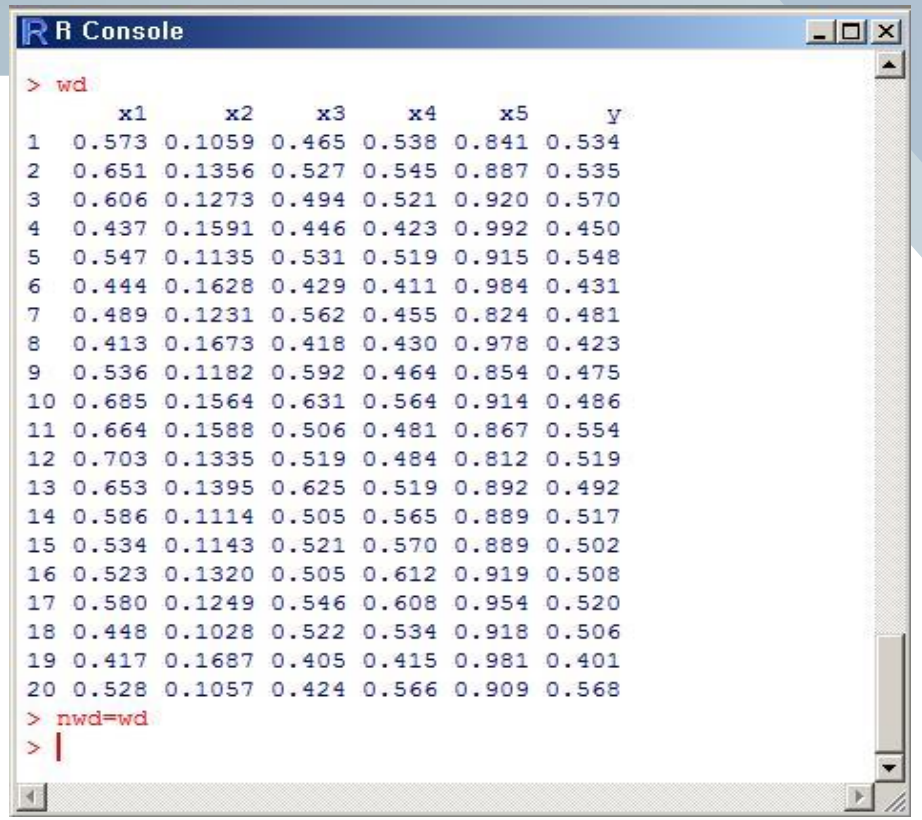
```
> cat("What is ₩"R₩"?₩n")
```

What is "R"?

7. 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  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
```



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

8. 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 of
different 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"
```

8. 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
```

03

R 데이터구조

1. Matrices and Arrays

◆ Matrix is a two-dimensional array 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
```

1. Matrices and Arrays

◆ 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
```

2. 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“
```

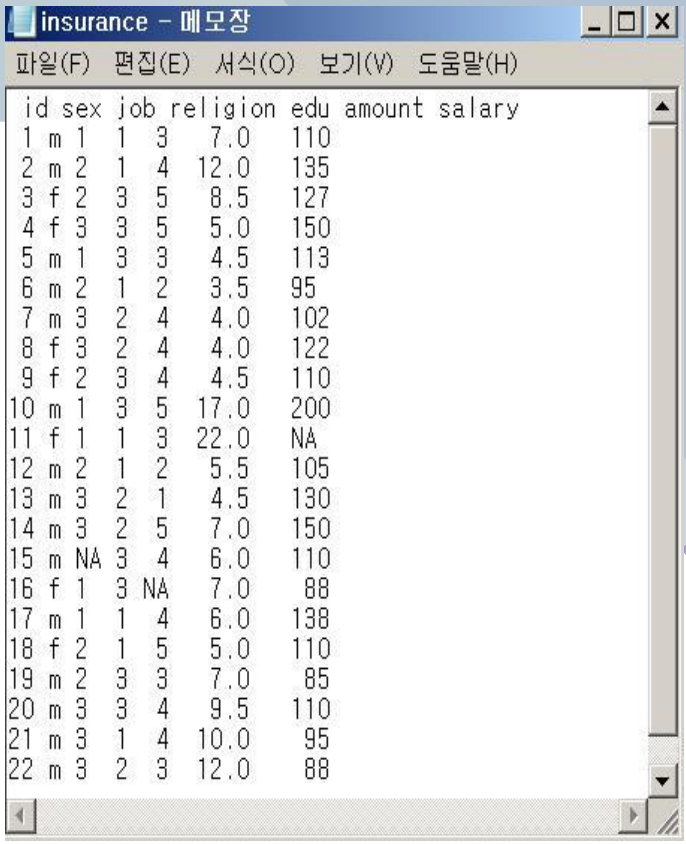
2. Factors

- 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:/grad/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

3. 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
```

4. 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
```

5. 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
```

04

R 데이터 편집

1. 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]  # conditional selection
[1] 3885 5160
> d[d$intake.pre > 6000,]   # conditional selection
  intake.pre intake.post
4       6180        5160
5       6390        5645
```

2. 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
```

다음시간 안내

03

Probability and Distributions