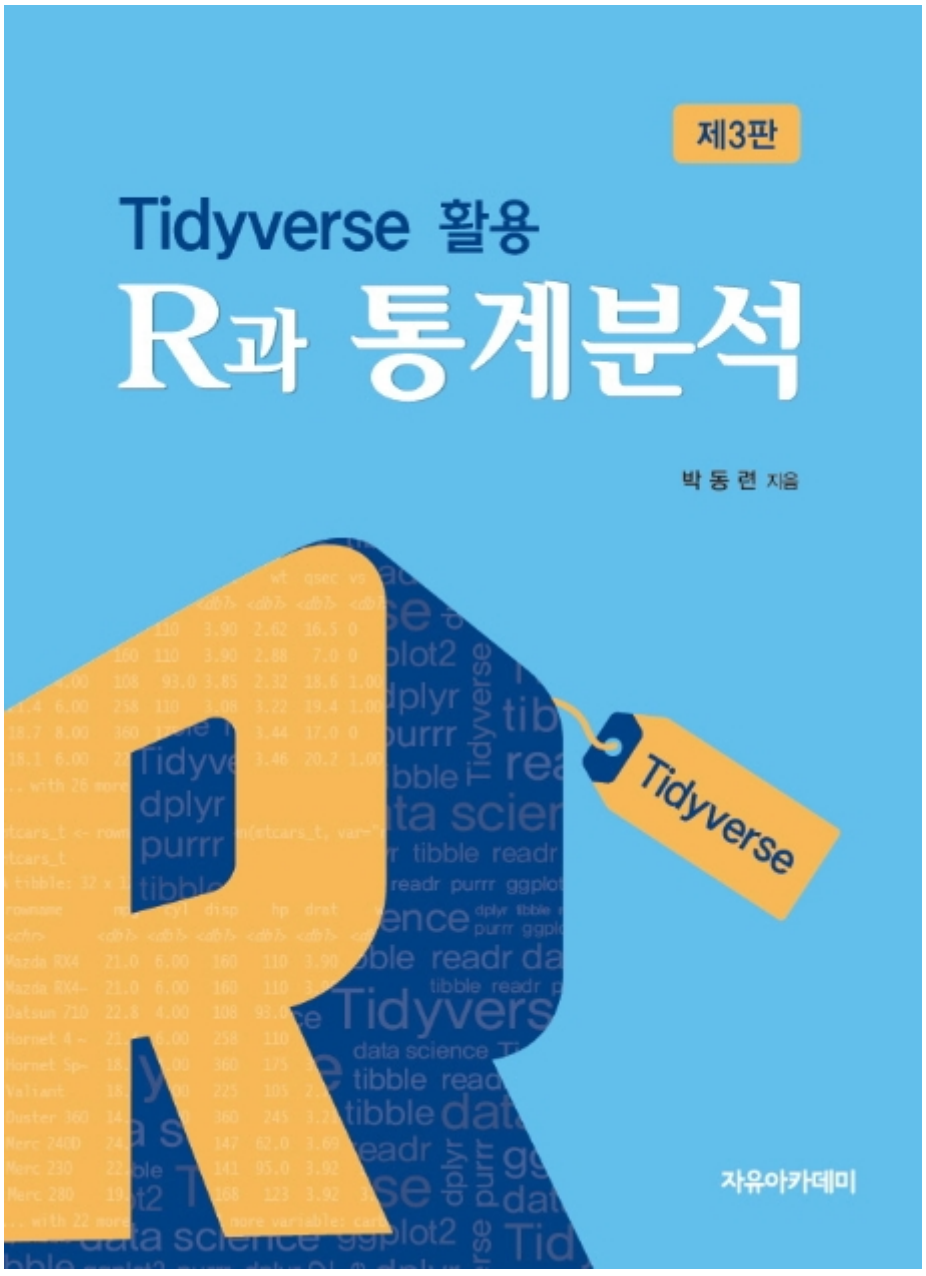
R

2022-02-11

Contents


제3판
Tidyverse 활용
R과 통계분석
박 동 련 지음
Tidyverse
자유아카데미

Chapter 1

R

Placeholder

1.1 R

1.2 R

1.3 RStudio R

1.4

1.5

1.6

1.7 R :

1.7.1

1.7.2

1.7.3 tidyverse

Chapter 2

R

. . ,
. 2 . ,
.

. R , , , , .
1 , , 2 , 2 .
, . .

2.1

1 . , ,
(logical), (numeric), (character) . (integer) (double) .

2.1.1

-

  c() .

```
> x <- c(TRUE, FALSE, TRUE)
> y1 <- c(1L, 3L, 5L)
> y2 <- c(1.1, 3.5, 10.4)
> z <- c("one", "two", "three")
```

x , y1 , y2 , z . TRUE FALSE , .
y1 L . , "
" ' ' .

-

typeof() .

```
> typeof(x)
[1] "logical"
> typeof(y1)
[1] "integer"
> typeof(y2)
[1] "double"
> typeof(z)
[1] "character"
```

, length() . length(y2) y2
3 .

(scalar) , c() . , a <- 1 a <- c(1)
a .

, ? .

```
> c(1, "1", TRUE)
[1] "1"    "1"    "TRUE"
> c(3, TRUE, FALSE)
[1] 3 1 0
```

1 , "1" , TRUE . .
. . , TRUE 1 , FALSE 0 .
.

-

. , , , ( ) .

```
> c(Seoul=9930, Busan=3497, Inchon=2944, Suwon=1194)
 Seoul  Busan Inchon  Suwon
  9930   3497   2944   1194
```

names() .

```
> pop <- c(9930,3497,2944,1194)
> names(pop) <- c("Seoul", "Busan", "Inchon", "Suwon")
> pop
 Seoul  Busan Inchon  Suwon
  9930   3497   2944   1194
```

- scan()

scan() . scan()
. Console > 1: ,
, . 4 5: Enter
.

```
> x <- scan()
1: 24
2: 35
3: 28 21
5:
Read 4 items
> x
[1] 24 35 28 21
```

what="character" . c() ,
.

```
> y <- scan(what = "character")
1: Seoul  Suwon
3: 'New York'
4:
Read 3 items
> y
[1] "Seoul"    "Suwon"    "New York"
```

scan() . .
D:\Data data1.txt . z scan()
.

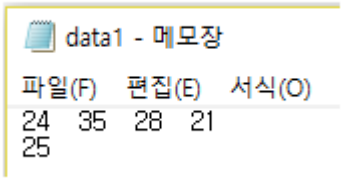


Figure 2.1: data1.txt

```
> z <- scan("Data/data1.txt")
> z
[1] 24 35 28 21 25
```

2.1.2

.

2.1.2.1

c() append() . c() .

```
> x <- c(11,12,13,14)
> c(x, 15)
[1] 11 12 13 14 15
> y <- c(16,17,18)
> c(x, y)
[1] 11 12 13 14 16 17 18
```

append() after .

```
> append(x, 15)
[1] 11 12 13 14 15
> append(x, 15, after=2)
[1] 11 12 15 13 14
> append(x, y)
[1] 11 12 13 14 16 17 18
> append(x, y, after=3)
[1] 11 12 13 16 17 18 14
```

2.1.2.2

. 1 100 . c()
. (:) seq() , rep() .

-

1 . 1 .

```
> 1:5
[1] 1 2 3 4 5
> -3:3
[1] -3 -2 -1  0  1  2  3
> 1.5:5.4
[1] 1.5 2.5 3.5 4.5
> 5:0
[1] 5 4 3 2 1 0
```

- seq()

1 . seq() .
from to by length . (from) (to) , 1 .
by from by ( ) to . length
from to . to , from by length .

```
> seq(from=0, to=5)
[1] 0 1 2 3 4 5
> seq(from=0, to=5, by=2)
[1] 0 2 4
> seq(from=0, to=5, length=3)
[1] 0.0 2.5 5.0
> seq(from=0, by=2, length=3)
[1] 0 2 4
```

seq() 1 1 ( ) .

```
> seq(3)
[1] 1 2 3
> seq(-3)
[1]  1  0 -1 -2 -3
```

, 1 1 . seq() along , length
. seq_along() seq_len() .

```
> x <- c(24,31,29)
> seq(along=x)
[1] 1 2 3
> seq_along(x)
[1] 1 2 3
> seq(length=length(x))
[1] 1 2 3
> seq_len(length(x))
[1] 1 2 3
```

- rep()

rep() times each, length . times
.

```
> rep(1, times=3)
[1] 1 1 1
> rep(1:3, times=2)
```

```
[1] 1 2 3 1 2 3
> rep(c("M","F"), times=c(2,3))
[1] "M" "M" "F" "F" "F"
```

times . c("M","F") c(2,3)
M F .

each , each . times each times .

```
> rep(1:3, each=2)
[1] 1 1 2 2 3 3
> rep(1:3, times=rep(2,3))
[1] 1 1 2 2 3 3
> rep(1:3, each=2, times=2)
 [1] 1 1 2 2 3 3 1 1 2 2 3 3
```

length , length . each , each
length .

```
> rep(1:3, length=6)
[1] 1 2 3 1 2 3
> rep(1:3, each=2, length=8)
[1] 1 1 2 2 3 3 1 1
```

2.1.3

R .
Table 2.1 .

Table 2.1:

nchar(x)	x
paste(..., sep=" ")	
substr(x, start, stop)	
toupper(x)	
tolower(x)	
strsplit(x, split)	
sub(old, new, x)	
gsub(old, new, x)	

- nchar():

nchar() . x 4 , 3 , 4 .
.

```
> x <- c("Park","Lee","Kwon")
> nchar(x)
[1] 4 3 4
> nchar("    ")
[1] 6
```

- paste():

. paste() . paste()
, sep .
5 “ ”, “ ”, “ ”, “ ” “ ” . sep sep=" "
, sep="-" (dash) , sep="" .

```
> paste(" ", "   ", "  ", "   ", "   ")
[1] "               "
> paste(" ", "   ", "  ", "   ", "   ",
+       sep="-")
[1] " -   - -   -  "
> paste(" ", "   ", "  ", "   ", "   ",
+       sep="")
[1] "           "
```

paste() .

```
> paste("  ", pi, " ")
[1] "   3.14159265358979  "
```

paste() . , . , "Stat"
1:3 c("Stat","Stat","Stat") 1:3 , c("Stat","Math")
1:3 , c("Stat","Math", "Stat") 1:3 .

```
> paste(c("Stat", "Math"), 1:2, sep = "")
[1] "Stat1" "Math2"
> paste("Stat", 1:3, sep="")
[1] "Stat1" "Stat2" "Stat3"
> paste(c("Stat","Math"), 1:3, sep="-")
[1] "Stat-1" "Math-2" "Stat-3"
```

paste() sep="" paste0() .

```
> paste0("stat", 1:3)
[1] "stat1" "stat2" "stat3"
```

. collapse .
letters LETTERS .

```
> letters
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
[20] "t" "u" "v" "w" "x" "y" "z"
> paste0(letters, collapse="")
[1] "abcdefghijklmnopqrstuvwxyz"
> paste(LETTERS, collapse=",")
[1] "A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"
```

collapse .

paste() , collapse ,
.

```
> paste0(letters, LETTERS, collapse = ",")
[1] "aA,bB,cC,dD,eE,fF,gG,hH,iI,jJ,kK,lL,mM,nN,oO,pP,qQ,rR,sS,tT,uU,vV,wW,xX,yY,zZ"
> paste(letters, LETTERS, sep = "-", collapse = ",")
[1] "a-A,b-B,c-C,d-D,e-E,f-F,g-G,h-H,i-I,j-J,k-K,l-L,m-M,n-N,o-O,p-P,q-Q,r-R,s-S,t-T,u-
```

- substr():

. substr() , , .
.

, . x
. (=1, =2), (=3, =6),
(=1, =2) .

```
> substr("Statistics", 1, 4)
[1] "Stat"
> x <- c("    ", "    ", "   ")
> substr(x, 3, 6)
[1] "  " "  " "  "
> substr(x, c(1,3), c(2,6))
[1] " "    "  " " "
```

x .
nchar() .

```
> x <- c("New York, NY","Ann Arbor, MI","Chicago, IL")
> substr(x, start=nchar(x)-1, stop=nchar(x))
[1] "NY" "MI" "IL"
```

ggplot2 mpg . 11 trans . trans . library() :: . mpg$trans mpg trans ?? . table() .

```
> x <- ggplot2::mpg$trans
> table(x)
x
  auto(av)   auto(l3)   auto(l4)   auto(l5)   auto(l6)   auto(s4)   auto(s5)
         5          2         83         39          6          3          3
  auto(s6) manual(m5) manual(m6)
        16         58         19
```

auto(av) auto(s6) auto manual(m5) manual(m6) manual 1, nchar(x)-4 .

```
> y <- substr(x, start=1, stop=nchar(x)-4)
> table(y)
y
  auto manual
   157     77
```

- strsplit():

. strsplit() split .

x . split="," . . R 2.7 .

```
> x <- c("New York, NY","Ann Arbor, MI","Chicago, IL")
> (y <- strsplit(x,split=","))
[[1]]
[1] "New York" " NY"

[[2]]
[1] "Ann Arbor" " MI"

[[3]]
[1] "Chicago" " IL"
> unlist(y)
[1] "New York"  " NY"       "Ann Arbor" " MI"       "Chicago"   " IL"
```

y , . unlist() , .

, split = "" .

```
> unlist(strsplit("PARK",split=""))
[1] "P" "A" "R" "K"
```

(.) split="." . split (regular expression) , . . **??** .

```
> unlist(strsplit("a.b.c",split="."))
[1] "" "" "" "" ""
> unlist(strsplit("a.b.c",split="[.]"))
[1] "a" "b" "c"
```

- toupper() tolower()

. toupper() tolower() .

```
> x <- c("park","lee","kwon")
> (y <- toupper(x))
[1] "PARK" "LEE"  "KWON"
> tolower(y)
[1] "park" "lee"  "kwon"
```

. x . substr() , toupper() , substr() .

```
> x
[1] "park" "lee"  "kwon"
> substr(x,1,1) <- toupper(substr(x,1,1))
> x
[1] "Park" "Lee"  "Kwon"
```

- sub() gsub():

. sub(old, new, x) gsub(old, new, x) , x old new . sub() old new , gsub() old new .

```
> x <- "Park hates stats. He hates math, too."
> sub("hat","lov",x)
[1] "Park loves stats. He hates math, too."
> gsub("hat","lov",x)
[1] "Park loves stats. He loves math, too."
```

.

```
> (y <- paste0("banana",1:3))
[1] "banana1" "banana2" "banana3"
> sub("a","A",y)
[1] "bAnana1" "bAnana2" "bAnana3"
> gsub("a","A",y)
[1] "bAnAnA1" "bAnAnA2" "bAnAnA3"
```

new "" .

```
> z <- "Everybody cannot do it"
> sub("not","",z)
[1] "Everybody can do it"
```

2.1.4

, . .

```
> x <- c(7,8,9,10)
> y <- c(1,2,3,4)
> x+y
[1]  8 10 12 14
> x-y
[1] 6 6 6 6
> x*y
[1]  7 16 27 40
> x/y
[1] 7.0 4.0 3.0 2.5
> x^y
[1]     7    64   729 10000
```

, , .

.

```
> x
[1]  7  8  9 10
> x+3
[1] 10 11 12 13
> x/4
[1] 1.75 2.00 2.25 2.50
> 2^x
[1]  128  256  512 1024
```

R . R (loop)
.

Inf, -Inf, NaN . Inf -Inf NaN 'Not a Number' .

```
> c(-1,0,1)/0
[1] -Inf  NaN  Inf
```

NaN 0/0 .

```
> sqrt(-1)
Warning in sqrt(-1): NaN
[1] NaN
> Inf-Inf
[1] NaN
> Inf/Inf
[1] NaN
```

-

R , ?
. 1:6 + 1:3 3 1:3 6 c(1,2,3,1,2,3)
.

```
> 1:6 + 1:3
[1] 2 4 6 5 7 9
> 1:6 + rep(1:3,2)
[1] 2 4 6 5 7 9
```

, .
. c(7,8,9,10) 3 3 4 c(3,3,3,3) .
. R .

```
> 1:4 + 1:3
Warning in 1:4 + 1:3:
[1] 2 4 6 5
```

. .

•

R . . Table 2.2 .

Table 2.2:

abs(x)		
sqrt(x)		
ceiling(x)	x	
floor(x)	x	
trunc(x)	x	
round(x, n)	x	n
signif(x, n)	x	n
log(x)	x	
log10(x)	x	
exp(x)	x	

```
> abs(-2)
[1] 2
> sqrt(25)
[1] 5
> ceiling(3.475)
[1] 4
> floor(3.475)
[1] 3
> trunc(5.99)
[1] 5
> round(3.475,2)
[1] 3.48
> signif(0.00347, 2)
[1] 0.0035
> sin(1); cos(1); tan(1)
[1] 0.841471
[1] 0.5403023
[1] 1.557408
> asin(sin(1)); acos(cos(1)); atan(tan(1))
```

```
[1] 1
[1] 1
[1] 1
> log(2,base=2)
[1] 1
> log(10)
[1] 2.302585
> log10(10)
[1] 1
> exp(log(10))
[1] 10
```

-

. Table 2.3 .

Table 2.3:

mean(x)	
median(x)	
range(x)	
IQR(x)	
sum(x)	
diff(x, n)	: x[i + n] - x[i], n 1
min(x)	
max(x)	

```
> x <- c(1,2,3,4,50)
> mean(x)
[1] 12
> median(x)
[1] 3
> range(x)
[1]  1 50
> IQR(x)
[1] 2
> sd(x)
[1] 21.27205
> var(x)
[1] 452.5
> sum(x)
[1] 60
```

```
> min(x)
[1] 1
> max(x)
[1] 50
> diff(c(1,2,4,7,11))
[1] 1 2 3 4
```

-

. R NA (not available) .
, is.na() .

```
> x <- c(1,0,3,5,NA)
> is.na(x)
[1] FALSE FALSE FALSE FALSE  TRUE
```

is.na() sum() .

```
> sum(is.na(x))
[1] 1
```

x == NA . x == NA
NA . == 2.1.5 .

NA NA . R NA
. NA na.rm=TRUE .

```
> mean(x)
[1] NA
> max(x)
[1] NA
> mean(x,na.rm=TRUE)
[1] 2.25
> max(x,na.rm=TRUE)
[1] 5
```

2.1.5

, .
.

Table 2.4: /

<	
<=	
>	
>=	
==	
!=	
!x	x (NOT)
x | y	x y (OR)
x & y	x y (AND)

R . , . 3
3 TRUE FALSE .

```
> x <- c(3,8,2)
> y <- c(5,4,2)
> x > y
[1] FALSE  TRUE FALSE
> x >= y
[1] FALSE  TRUE  TRUE
> x < y
[1]  TRUE FALSE FALSE
> x <= y
[1]  TRUE FALSE  TRUE
> x == y
[1] FALSE FALSE  TRUE
> x != y
[1]  TRUE  TRUE FALSE
```

.

```
> x <- 1:3
> x > 2
[1] FALSE FALSE  TRUE
> x < 2
[1]  TRUE FALSE FALSE
> x <= 2 | x >= 3
[1] TRUE TRUE TRUE
> x <= 2 & x >= 1
[1]  TRUE  TRUE FALSE
```

. ,
any() all() .

```
> x <- 1:5
> any(x>=4)
[1] TRUE
> all(x>=4)
[1] FALSE
```

. x 4
sum() mean() .

```
> x <- 1:5
> x >= 4
[1] FALSE FALSE FALSE  TRUE  TRUE
> sum(x>=4)
[1] 2
> mean(x>=4)
[1] 0.4
```

%in% .

```
> x <- 1:5
> x %in% c(2,4)
[1] FALSE  TRUE FALSE  TRUE FALSE
```

x %in% TRUE, FALSE .

x c(2,4) x == c(2,4) .

```
> x == c(2,4)
Warning in x == c(2, 4):
[1] FALSE FALSE FALSE  TRUE FALSE
```

x c(2,4) . c(2,4) x == c(2,4,2,4,2) , .

2.1.6

(indexing) , . , x[a] , a , .

, . .

```
> y <- c(2,4,6,8,10)
> y[c(1,3,5)]
[1]  2  6 10
> y[c(-2,-4)]
[1]  2  6 10
```

, NA .

```
> y[c(2,2,2)]
[1] 4 4 4
> y[0]
numeric(0)
> y[6]
[1] NA
```

. pop Seoul Suwon .

```
> pop <- c(Seoul=9930, Busan=3497, Inchon=2944, Suwon=1194)
> pop[c("Seoul", "Suwon")]
Seoul Suwon
 9930  1194
```

TRUE , .

```
> y
[1]  2  4  6  8 10
> y[c(TRUE,TRUE,FALSE,FALSE,TRUE)]
[1]  2  4 10
> y>3
[1] FALSE  TRUE  TRUE  TRUE  TRUE
> y[y>3]
[1]  4  6  8 10
```

-

TRUE , .
. x x . x .

```
> x <- c(80,88,90,93,95,94,99,78,101)
> x >= mean(x)
[1] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
```

FALSE , TRUE . .

```
> x[x >= mean(x)]
[1]  93  95  94  99 101
```

.

1) ± 1

2) ± 1 ± 2

3) ± 2

x z z x . ± 2
.

```
> z <- (x-mean(x))/sd(x)
> x[abs(z) <= 1]                # 1
[1] 88 90 93 95 94
> x[abs(z) > 1 & abs(z) <= 2]   # 2
[1]  80  99  78 101
> x[abs(z) > 2]                 # 3
numeric(0)
```

2.2

2.2.1

2.2.2

2.3

2.4

2.4.1

2.4.2

2.5

2.5.1

2.5.2

2.5.3 `with()`

2.6 Tibble:

2.6.1 Tibble

2.6.2 Tibble

2.7

2.8

1. `iris` `setosa` , `versicolor` , `virginica` . 50 50 `setosa`, 51 100 `versicolor`, 50 `viginica` .

- `iris` 1~3 51~53 101~103 .

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
101	6.3	3.3	6.0	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica
103	7.1	3.0	5.9	2.1	virginica

- iris Sepal.Length, Sepal.Width, Petal.Length, Petal.Width .
- 150 Petal.Width 1 Petal.Length 4 .

2. mtcars 1974 32 .

- mpg grade . , $\bar{x}$ sd mpg .

$mpg \leq \bar{x} - sd$	grade = "Bad"
$\bar{x} - sd < mpg \leq \bar{x} + sd$	grade = "Good"
$mpg > \bar{x} + sd$	grade = "Excellent"

- mtcars model
- grade Excellent model mpg .
- grade Bad mpg .

Chapter 3

Placeholder

3.1 : readr

3.1.1 read_table()

3.1.2 read_csv() CSV

3.1.3 read_fwf()

3.2 Excel

3.3 SAS

3.4 HTML

Chapter 4

dplyr

Placeholder

4.1

4.1.1 : filter()

4.1.2 : slice()

4.1.3 : arrange()

4.1.4 : distinct()

4.2

4.2.1 : select()

4.2.2 : rename() rename_with()

4.2.3 : relocate()

4.2.4 : mutate() transmute()

4.3 : summarise()