

2.1 R 소개

2.1.1 단순계산

```
2+3
2*3
2-3
2/3
2^3
3-2*1+4
2+3 ; 2*3 ; 2-3 ; 2/3 ; 2^3      # ; 한줄에 여러 명령어 입력 때
(20+3*2)/3                        # 소숫점 7자리에서 반올림
options(digits=4) ; (20+3*2)/3     # 앞으로 계속 소숫점 4자리에서 반올림
```

2.1.2 함수계산

```
pi ; PI                          # 3.14..., 대문자와 소문자를 구분(Error 발생)
sqrt(2) ; sqrt(100)             #  $\sqrt{\quad}$  계산
sin(pi) ; cos(pi/2) ; tan(pi/4) ; sin(3.14) ; cos(1) # 삼각함수 계산(라디안)
# > sin(pi)
# [1] 1.224606e-16
# sin(pi)는 원래 "0"입니다. 그런데  $\pi$ 는 무리수라서 정확한 값을 구할 수 없는 무리수입니다. 그래서 계산결과는  $1.224606 \times 10^{-16}$ 으로 거의 0입니다.(e 가 10입니다.)
asin(1) ; acos(1) ; atan(1)      # arcsine, arccosine, arctan 삼각함수의 역함수
exp(1) ; exp(10) ; exp(100) ; exp(500) ; exp(1000) #  $e^x$ 
log(10) ; log(e) ; log(exp(1)) ; log(exp(100))     # log : 자연로그, 밑이 e
log(10, base=10) ; log(100, base=10) ; log(2, base=10) #  $\log_{10} x$  상용로그
log(10, base=3) ; log(100, base=3) ; log(2, base=3)  #  $\log_3 x$ 
abs(-3) ; abs(3)                                     # absolute value (절대치)
factorial(5) ; factorial(10) ; factorial(100) ; factorial(200) # n!
> factorial(5) ; factorial(10) ; factorial(100) ; factorial(200)
[1] 120
[1] 3628800
[1] 9.332622e+157
[1] Inf
경고메시지(들):
In factorial(200) : value out of range in 'gammafn'
```

100!은 9.332622×10^{157} 로 엄청나게 큰수 이지요.

200!은 inf : 컴퓨터에서 표현할 수 있는 수를 넘는 수, 즉 무한대라는 겁니다.

그럼 컴퓨터에서 표현할 수 있는 수의 최대치는 얼마인지 한번 조사해 보시길...

```
choose(5,2) ; choose(10,2) ; choose(100,50) ; choose(1000,500) #  ${}_nC_r$ 
```

여기서도 컴퓨터에서 표현할 수 있는 수의 최대치는 얼마인지 조사해 보시길...

${}_nP_r = {}_nC_r \times n!$

2.1.3 경고(Error)이 발생하는 겁니다. Error확인하고 고칠수 있는거는 고쳐보시길)

```
log(e)           # log (exp(1))
square(3)        # 3^2
squareroot(3) ; sqrt{3}      #  $\sqrt{3}$  sqrt(3)
sqrt(-3)         #  $\sqrt{-3}$  은 실수에서는 정의가 되지 않습니다.
a                # a 라는 변수에 아직 값이 없습니다.
```

2.2 할당 (인제 바로 위의 a가 값을 갖게 할당을 배웁니다.)

```
x1 = 3 ; x1      # = x1에 3을 할당했습니다. 인제 x1 을 치면 3으로 나옵니다.
x2 <- 9 ; x2     # = 와 <- 는 같은 기능, x2에 9할당
x1 + x2 ; x1-x2 ; x1*x2 ; x2/x1 ; x2^x1 # 어떤 결과가 나올까요?
x=3 ; x+4       # x에 3을 할당
e               # e 에는 여러분 컴퓨터에는 할당이 없습니다. 그래서 Error..
e <- exp(1) ; e  # 인제 e 에 e 값을 할당했습니다.
y <- e^2 ; y     # y에  $e^2$ 을 할당 했습니다.
z = x + y ; z    # z에 뭐가 할당 되었을까요??
```

2.3 벡터만들기 (c(, ,) 형식입니다. c는 열 또는 세로를 뜻하는 column 의 첫 자계지요.)

2.3.1 숫자형 - 통계에서 많이 사용(통계에서 사용하는 자료는 전부 벡터형식입니다.)

```
k_score = c(96, 80, 76, 96, 88, 75, 78, 89, 92, 70) ; k_score
k_score_2 = c(67, 83, 96, 90, 85, 75, 82, 89, 92, 75) ; k_score_2
score=c(k_score, k_score_2) ; score #score에는 뭐가 할당 될까요?

x = c(1,2,3,4,5) ; y = c(-1,-2,-3,-4,-5) ; x ; y
x[1] ; x[3] ; x[5] ; x[7]      # x[3]은 x벡터의 3번째 수, 통계량 계산시 중요
y[1] ; y[3] ; y[5] ; y[7]
어떤 의미인지 결과를 보고 이해해 보십시오..
```

2.3.2 문자형

```
family = c("song", "kim", "hee", "yoon") #우리 가족입니다.
names(family) = c("father", "mother", "son1", "son2") #전 들들이 아빠 ㅋㅋ
family
```

2.3.3 논리형

```
x = c(T, F, F, F, T, T, F) ; x
sum(x)                        # x에서 TRUE의 갯수
y = c(1, 3, 5, 7, 9, 10, 100)
z = y < 8 ; z                 # y 벡터 중에 8보다 작은 것은 TRUE, 아니면 F
sum(z)                        # y에서 TRUE의 갯수
```

2.4 벡터 함수

2.4.1 열벡터에서의 기초통계량

```
k_score
sum(k_score) ; mean(k_score) ; median(k_score) #합, 평균, 중앙값
max(k_score) ; min(k_score)
range(k_score)
var(k_score) ; sd(k_score)
length(k_score) # 열의 길이, 반복문에서 자주 사용
```

2.5 구조적 데이터(중요)

일정한 패턴을 가지는 자료입니다, 어떤 구조적인지 실행하면서 잘 이해해 보십시오.
자주 사용됩니다.

```
1:10 ; seq(1:10) ; seq(1,10) # 오름차순으로
10:1 ; rev(1:10) # 내림차순으로
seq(1,10, by = 2) ; seq(1,10, by = 0.5) # 해보시면 압니다
seq(from = 5, to=22, by = 2)
rep(1,10) ; rep(1:3,3) # 반복되는 원소
rep(c(2,3,5), 4)
a = rep(0,12) ; a # 초기값을 0으로 두고 프로그래밍에 사용
```

R에 내장된 자료(아무래도 미국 중심이겠지요...)

```
LETTERS[1:10] # 연속되는 대문자 알파벳
letters[1:10] # 연속되는 소문자 알파벳
month.name # 해보시면 압니다.
month.abb
state.name
```

[과제3] (수업시간에 한 내용 모두 + 각 절에서 비슷한 내용으로 한번 더 해 보시길..)

첨부파일 : 학번이름3.hwp (예 : 20192260홍길동3.hwp)

- R console 창에서의 프로그램, 결과
- 결과해석 순으로...

(이번에는 결과 밑에 간단히 마음껏(맘대로) 해석을 달아 보십시오)

주어진 시간 안에 꼭 과제를 제출하시기 바랍니다.

다음 페이지부터는 벡터연산과 행렬 연산이 있습니다.

그런데 통계학과 2학년들은 아직 벡터와 행렬을 배우지 않았습니다.

그래서 생각합니다만....

혹시 필요한 학생은 참고 하시기..

[NOTE] 이후로는 생략합니다.-(안해도 된다는 말입니다)

참조하신분만 하시기....

2학년은 아직 벡터와 행렬을 안배웠네요...

통계수학2를 수강하고 참고 하세요.

2.4.2 벡터 연산

```
x = c(1,2,3,4,5) ; x
y = c(-1,-2,-3,-4,-5) ; y
t(x) ; x+y ; x-y      # t(x) : 벡터 x의 transposed vector (전치벡터)
x*y ;                  # 벡터곱 아님
x/y                    # 벡터 나누기 아님

t(x) %%% y            ## 벡터곱
x %%% t(x)
```

2.6 행렬 연산

```
a = c(1,3,5) ; a
b = matrix(1:6, nrow = 2) ; b
d = matrix(1:6, ncol = 2) ; d

a1 = c(1,2) ; a2 = c(3,4) ; a3 = c(5,6)
am = cbind(a1,a2,a3) ; am
```

2.6.1 기본 연산

```
x = c(1,2,3,4,5) ; y = c(-1,-2,-3,-4,-5)
x ; y
t(x) ; x+y ; x-y
x*y ; x/y

t(x) %%% y
x %%% t(x)

temp = c(x,y) ; temp
temp1 = cbind(x,y) ; temp1
temp2 = rbind(x,y) ; temp2

temp1 * temp1
t(temp1)
t(temp1) %%% temp1
dim(temp1)

a1 = c(1,2,3) ; a2 = c(4,5,6) ; a3 = c(7,8,9)
b1 = c(11,12,13) ; b2 = c(14,15,16) ; b3 = c(17,18,19)
A = rbind(a1, a2, a3) ; A # 3*3 행렬 A 생성 (행 기준)
B = rbind(b1, b2, b3) ; B # 3*3 행렬 B 생성 (행 기준)

A * B
A %%% B
B %%% A
```

2.6.2 역행렬

```
a1 = c(1,2) ; a1
```

```
a2 = c(3,4) ; a2
```

```
a = cbind(a1,a2) ; a
```

```
b = ginv(a) ; b
```

```
library(MASS) ✓
```

```
b = ginv(a) ; b
```

```
a %*% b ; b %*% a
```

```
IA = ginv(A) ; IA
```

```
A %*% IA
```

```
a1 = c(8,2,6) ; a2 = c(7,15,52) ; a3 = c(3,7,9)
```

```
A = rbind(a1, a2, a3) ; A
```

```
IA = ginv(A) ; IA
```

```
A %*% IA
```

2.7 연립방정식의 해

2.7.1 일차 연립방정식의 해

$$8x_1 + 2x_2 + 6x_3 = 1$$

$$7x_1 + 15x_2 + 52x_3 = 3$$

$$3x_1 + 7x_2 + 9x_3 = 10$$

```
a1 = c(8,2,6)
```

```
a2 = c(7,15,52)
```

```
a3 = c(3,7,9)
```

```
b = c(1,3,10)
```

```
A = rbind(a1, a2, a3) ; A
```

```
IA = ginv(A) ;
```

```
x = IA %*% b ; x
```

2.7.2 고유값, 고유벡터

```
eigen(A) # 고유값과 고유벡터 계산 - 다변량에서 필요
```