

제2강

변수와 벡터

Section 01

R의 기본 연산

1. R의 기본 연산

1. 산술연산과 주석

코드 2-1

```
2+3  
(3+6)*8  
2^3
```

2의 세제곱

```
> 2+3  
[1] 5  
> (3+6)*8  
[1] 72  
> 2^3
```

2의 세제곱

```
[1] 8
```

- 일반적으로 R에서는 한 줄에 하나의 명령문을 입력한다.
- 한 줄 내에서 # 이후의 내용은 주석으로 간주하여 실행하지 않는다.

1. R의 기본 연산

연산자	의미	사용 예
+	덧셈	3+5+8
-	뺄셈	9-3
*	곱셈	7*5
/	나눗셈	8/3
%%	나눗셈의 나머지	8%%3
^	제곱	2^3

표 2-1 산술연산자

코드 2-2

```
7+4  
# 2^3
```

```
> # 2^3
```

1. R의 기본 연산

2. 산술연산 함수

코드 2-3

```
log(10)+5      # 로그함수  
sqrt(25)       # 제곱근  
max(5,3,2)     # 가장 큰 값
```

```
> log(10)+5      # 로그함수  
[1] 7.302585  
> sqrt(25)       # 제곱근  
[1] 5  
> max(5,3,2)     # 가장 큰 값  
[1] 5
```



1. R의 기본 연산

함수	의미	사용 예
log()	로그함수	log(10), log(10, base=2)
sqrt()	제곱근	sqrt(36)
max()	가장 큰 값	max(3,9,5)
min()	가장 작은 값	min(3,9,5)
abs()	절대값	abs(-10)
factorial()	팩토리얼	factorial(5)
sin(), cos(), tan()	삼각함수	sin(pi/2)

표 2-2 함수

Section 02

변수

2. 변수

코드 2-4

```
a <- 10  
b <- 20  
c <- a+b  
print(c)
```

```
> a <- 10  
> b <- 20  
> c <- a+b  
> print(c)  
[1] 30
```


2. 변수

1. 변수의 개념

- 프로그램에서 어떤 값을 저장하는 저장소나 보관 박스

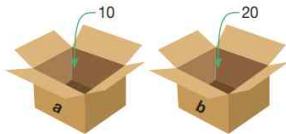


그림 2-1 변수의 개념: 변수, 변수명, 값

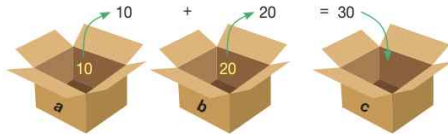


그림 2-2 $c \leftarrow a + b$ 의 실행 과정

2. 변수

- `a <- 10`

10을 변수 a에 저장

- `b <- 20`

20을 변수 b에 저장

- `c <- a+b`

변수 a의 값과 변수 b의 값을 더하여 변수 c에 저장

- `print(c)`

변수 c의 값을 출력('c'만 입력해도 출력)

2. 변수

2. 변수명 지정

- ① 첫 글자는 영문자(알파벳)나 마침표(.)로 시작하는데, 일반적으로 영문자로 시작

ex) avg, .avg

ex) 12th는 숫자로 시작했기 때문에 변수명 사용 불가

- ② 두 번째 글자부터는 영문자, 숫자, 마침표(.), 밑줄(_) 사용 가능

ex) v.1, a_sum, d10

ex) this-data, this@data은 변수명 사용 불가(@과 - 같은 특수문자 사용 불가)

- ③ 대문자와 소문자를 구분

ex) var_A 와 var_a는 서로 다른 변수

- ④ 변수명 중간에 빈칸을 넣을 수 없음

ex) first ds는 변수명 사용 불가

3. 변수에 값 저장 및 확인

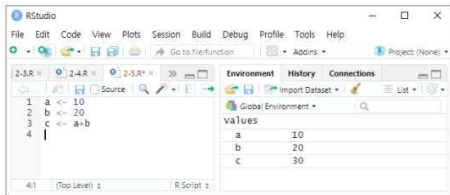
```
a <- 10      # 권장
b = 20      # 권장하지 않음
```

2. 변수

코드 2-5

```
a <- 125  
a  
print(a)
```

```
> a <- 125  
> a  
[1] 125  
> print(a)  
[1] 125
```



Alt와 - 를 누르면
'<-'가 입력

그림 2-3 R 스튜디오의 환경 창에서 변수의 내용 확인하기

2. 변수

4. 변수의 자료형

자료형	사용 예	비고
숫자형	1, 2, 3, -4, 12.8	정수와 실수 모두 가능
문자형	'Tom', "Jane"	작은 따옴표나 큰 따옴표로 묶어서 표현
논리형	TRUE, FALSE	반드시 따옴표가 없는 대문자로 표기하며, T나 F로 줄여서 사용하는 것도 가능
특수값	NULL	정의되어 있지 않음을 의미하며, 자료형도 없고 길이도 0임
	NA	결측값(missing value)
	NaN	수학적으로 정의가 불가능한 값 예 sqrt(-3)
	Inf, -Inf	양의 무한대(Inf), 음의 무한대(-Inf)

표 2-3 R에서 사용할 수 있는 값들의 자료형

2. 변수

5. 변수의 값 변경

- 변수에 저장된 값은 언제든지 변경 가능
- 변수의 자료형은 어떤 값을 저장하는가에 따라 유동적으로 바뀜

코드 2-6

```
a <- 10          # a에 숫자 저장
b <- 20
a+b             # a+b의 결과 출력
a <- "A"        # a에 문자 저장
a+b            # a+b의 결과 출력. 에러 발생
```

```
> a <- 10          # a에 숫자 저장
> b <- 20
> a+b             # a+b의 결과 출력
[1] 30
> a <- "A"        # a에 문자 저장
> a+b            # a+b의 결과 출력. 에러 발생
Error in a + b : non-numeric argument to binary operator
```

감사합니다.