

#### ##### 변수설정

# R 객체가 저장된 위치의 이름

## \* 변수명은 영문자, 한글, .로 시작하고 이후 숫자, \_ 사용가능

## \* 일부 예약어는 사용할 수 없음

## - 예) if, for, while, TRUE, FALSE ...

## \* 대문자와 소문자 구분

## - 예) If, For, While, True, False는 변수명으로 사용가능

#####

X1 = 1.0 (X1 = 1.0) : 양 옆으로 괄호를 씌우면 대입과 동시에 출력

1X = 1.0 사용 불가능한 변수명: 숫자로 시작

.X = 1.0

한글X = 1.0

if = 0 사용 불가능한 변수명: 예약어

If = 0

#### ##### Assign Operators

# =, <-, ->

## \* = : 오른쪽의 연산 값을 왼쪽의 변수에 할당(대입)

## \* <- : 오른쪽의 연산 값을 왼쪽의 변수에 할당(대입)

## \* -> : 왼쪽의 연산 값을 오른쪽의 변수에 할당(대입)

#####

a = 15

b <- 15; 5 -> c # 한줄에 여러 명령어를 쓸 때에는 ;로 분리

a <- b <- c <- 6 # 동일 연산자 사용 c, b, a 순서로 6이 들어감

(x <- 1/pi) # 대입결과를 출력하기 pi값이 미리 R에 지정돼있음

x <- c(1,2,3,4,5) # 'c()' 자료를 결합하여 벡터로 만드는 함수

# "R에서의 최소단위는 벡터"



a = b <- c <- 6 : 중간에 다른 연산자가 들어갈 경우 에러

#### ##### Arithmetic Operators

# +, -, \*, /, ^, \*\*, %% (나머지), %/% (몫)

## \* 연산순위: (^, \*\*) > (%%, %/%) > (\*, /) > (+, -)

## \* ( ) 안의 내용을 우선 연산

## \* 같은 우선 순위인 경우 왼쪽부터 먼저 실시

#####

n <- 10+2

3^3

n <- (3+5)\*3-4\*\*2/3 # 연산의 우선순위

58 %% 3

58 %/% 3

# ?: 홀수/짝수 구분 2로 나눴을 때 나머지로 구분

무한대(infinite)

x <- 0; y <- Inf # NA: Not Applicable/Available: 결측값(해당사항없음)

x\*y # NaN: Not a Number 정의할 수 없음

x/x # NaN

x <- c(1,2,3,4); y <- c(2,3,5,4) # 벡터연산

x+y # 3, 5, 8, 8 : 여러 개의 데이터가 있는 경우 그 데이터들의 순서가 같은 것끼리 연산

x <- c(1,2,3,4); y <- c(1,2) # 재활용규칙(recycling rule)

v <- x+y # 2, 4, 4, 6 다시 앞으로 간 것

x <- c(1,2,3,4,5)

w <- x+y # 2, 4, 4, 6, 6 (warning: 배수관계가 아님)

#### ##### Logical Operators

# !, &, &&, |, ||, xor(x, y)

## \* 결과는 TRUE, FALSE

## \* TRUE는 T, FALSE는 F로 표시할 수 있음

## \* TRUE는 T, FALSE는 F로 표시할 수 있음

## \* TRUE는 1, FALSE는 0으로 처리

## \* 0이면 FALSE, 0이 아닌 숫자(Inf 포함)는 TRUE로 처리

# &&, || : 결과가 여러 개인 경우 첫번째 것만 반영

#####

x = TRUE; y = c(FALSE, TRUE)

!y	T F
x & y	F T : recycle
x && y	F
x   y	T T
x    y	T
x + y	0 1
x * y	Inf 1
x / y	

x <- c(0,2,4,-3); !x    T F F F

x <- c(NA, TRUE); y <- c(FALSE, NA) #NA: Not Applicable/Available: 결측

값(해당사항없음) NA & T : NA에 의해 결정됨

NA | T : 무조건 T

NA & F : 무조건 F

x & y    F NA

x | y    NA T

NA | F : NA에 의해 결정됨

#### ##### Relational Operators

# >, <, >=, <=, ==, !=

## \* 결과는 TRUE, FALSE

#####

x <- c(1,2,3,-1)

x > 0    T T T F

x1 <- 0.1 ; x2 <- 0.1; x3 <- 0.5 - 0.4

x1 == x2    T

x1 == x3    F : 실수 계산의 한계

identical(x1,x3)    F : 실수 계산의 한계(모든 비트를 비교함)

all.equal(x1,x3)    T : 유효한 숫자로만 비교하는 함수

x <- 45; (x > 10)+(x > 20)+(x > 30)+(x > 40)    4

조건을 다르게 주어 연산 가능

: 10으로 나눴을 때 나머지를 구하는 것과 같은 연산