

데이터 유형

R에는 데이터 유형(type of data)이 있다. 즉, 데이터가 어떤 값으로 이루어져 있는가를 의미한다. 숫자로 되어 있는지, 문자로 되어 있는지 등을 말한다.

R에서 사용되는 데이터 유형에는 기본적인 데이터 유형(Numeric(숫자), Character(문자), Logical(논리))과 특수한 형태의 데이터 유형이 있다.

1) 기본적인 데이터 유형

(1) Numeric(수치형)

① 숫자로 구성된 데이터로 double(실수형), integer(정수형)이 있다.

② 특별히 지정을 해주지 않는다면 자동으로 numeric 으로 들어간다.

```
x1 <- pi
class(x1) ; class(pi) # 데이터가 어떤 유형인지 알고 싶은 경우 class(변수명 또는
데이터 값) 또는 mode(변수명 또는 데이터 값) 사용한다.1)
[1] "numeric"
is.numeric(x1) ; is.numeric(pi) ; is.integer(pi) # 데이터가 numeric유형이 맞는지
확인하는 경우 is.numeric( ) 사용한다.2)
[1] TRUE
[1] TRUE
[1] FALSE
x1 <- as.integer(x1) # 데이터 유형을 변경하고 싶은 경우 as.유형명( )을 사용한다.
class(x1)
[1] "integer"
```

③ numeric은 실수로 표현되고, integer는 정수만 표현 가능하다.

```
x <- -8
x/3
[1] -2.666667
as.integer(x/3) ; as.integer(-8/3)
```

1) 세미콜론(;)은 하나의 명령어가 끝났음을 알려주는 기능이다. 하지만 한 줄에 하나의 명령밖에 없으면 세미콜론을 해 주지 않아도 명령어가 끝났음을 인식한다.

2) 현재까지 할당된 모든 객체의 이름을 확인하려면 ls() 혹은 objects() 함수를 사용한다.

```
[1] -2      # 정수부분만 나타낸다.  
[1] -2
```

(2) Character(문자형)

- ① 텍스트로 구성되어 있으며 “Kim”, “Park” / "상", “중”, “하” / ‘남’ ‘여’와 같은 형태로 표시한다.
- ② 큰따옴표 혹은 작은 따옴표를 사용하면 문자형으로 취급해 저장한다.(숫자도 "-0.9"이나 '10'처럼 따옴표로 묶여 있으면 문자형이 된다.
- ③ 문자형은 대문자와 소문자를 구분하여 "data"와 "DATA"는 다르다고 본다.

```
x <- "Big data"  
class(x)  
[1] "character"  
  
x <- as.factor(x) # as.factor(x) : x의 데이터 유형을 factor로 변경  
class(x)  
[1] "factor"      # factor는 범주형 자료유형을 의미
```

(3) Logical(논리형)

- ① 논리형 타입은 TRUE와 FALSE 두 가지가 있다. 이때 TRUE, FALSE는 모든 글자를 대문자로 써야한다.
- ② TRUE, FALSE를 머리글자 T, F로 표시해도 무관하다.

```
x <- true  
Error: object 'true' not found  
x <- T  
x  
[1] TRUE
```

- ③ 숫자화시켜서 보자면 일반적으로 FALSE는 0을 TRUE는 1을 뜻한다.

```
x <- TRUE ; y <- FALSE
as.numeric(x) ; as.numeric(y)
[1] 1
[1] 0
```

2) 특수한 형태의 데이터 유형

(1) NULL

- ① NULL은 존재하지 않는 객체를 뜻하며, 변수가 초기화되지 않았을 때 사용한다.
- ② NULL은 프로그래밍의 편의를 위해 미정(undefined) 값을 표현하는 데 사용하는 개념이다.
- ③ 어떤 변수에 NULL이 저장되어 있는지는 **is.null()**을 사용해 판단할 수 있다.

is.null(x) # 변수에 NULL이 저장되어 있는지를 판단한다.

반환 값은 NULL이 저장되어 있으면 TRUE, 그렇지 않으면 FALSE다.

다음은 NULL을 변수에 저장하고 이를 **is.null()**로 확인하는 예다.

```
x <- NULL
is.null(x)
[1] TRUE
is.null(1)
[1] FALSE
is.null(NA)
[1] FALSE
```

(2) NA

- ① R과 다른 언어의 가장 큰 차이 중 하나가 바로 **NA**(Not Available) 상수다.
- ② NA는 데이터 값이 없음(결측치)을 뜻한다.

예를 들어, 4명의 시험 점수가 있을 때 3명의 점수는 각각 80, 90, 75지만 4번째 사람의 점수를 모를 경우 NA를 이용해 4번째 사람의 점수를 표현한다.

```
x1 <- 80
x2 <- 90
x3 <- 75
x4 <- NA
```

③ 변수에 NA 값이 저장되어 있는지는 **is.na()** 함수로 확인한다.

is.na(x) # x는 R의 데이터 객체

NA가 저장되어 있으면 TRUE, 그렇지 않으면 FALSE를 출력한다.

예를 들어, 다음은 변수 x4에 NA가 저장되어 있음을 확인하는 코드다.

```
is.na(x4)
[1] TRUE
x5 <- x1+x2+x3+x4
is.na(x5)
[1] TRUE
x <- c(x1, x2, x3, x4)
is.na(x)
[1] FALSE FALSE FALSE TRUE
```

(3) NaN Not Available Number 수학적으로 계산이 불가능한 경우

(4) Inf 무한대. Inf는 양의 무한대 -Inf 는 음의 무한대

3) 데이터 유형을 알려주는 **is**로 시작하는 함수들은 아래의 표와 같다.

함수명	설명	입력내용	결과내용
is.numeric()	수치형 여부	is.numeric(데이터)	TRUE or FALSE
is.integer()	정수형 여부	is.integer(데이터)	
is.double()	실수형 여부	is.double(데이터)	
is.character()	문자형 여부	is.character(데이터)	
is.logical()	논리형 여부	is.logical(데이터)	
is.complex()	복소수형 여부	is.complex(데이터)	
is.null()	NULL 여부	is.null(데이터)	
is.na()	NA 여부	is.na(데이터)	
is.finite()	유한수치 여부	is.finite(데이터)	
is.infinite()	무한수치 여부	is.infinite(데이터)	

4) 데이터의 유형을 강제로 변경하는 함수는 다음과 같다.

함수명	설명	입력내용	결과 내용
<code>as.numeric()</code>	수치형으로 변환	<code>as.numeric(데이터)</code>	변환되거나 NA
<code>as.integer()</code>	정수형으로 변환	<code>as.integer(데이터)</code>	
<code>as.double()</code>	실수형으로 변환	<code>as.double(데이터)</code>	
<code>as.character()</code>	문자형으로 변환	<code>as.character(데이터)</code>	
<code>as.logical()</code>	논리형으로 변환	<code>as.logical(데이터)</code>	
<code>as.complex()</code>	복소수형으로 변환	<code>as.complex(데이터)</code>	