

PR3 - Vector

김서준

2023년 9월 21일

1. R에서 기초적인 Data Type

1.1. Numeric : 숫자 데이터 인식, 정수, 실수 등

```
num <- 3 ; class(num)
```

```
## [1] "numeric"
```

```
numvec <- c(1, 2, 3) ; class(numvec)
```

```
## [1] "numeric"
```

1.2. Complex : 복소수 $a+bi$

```
comp <- 2 + 3i ; class(comp)
```

```
## [1] "complex"
```

```
compvec <- c(2 + 3i, 4 + 5i, 6 + 7i) ; class(compvec)
```

```
## [1] "complex"
```

1.3. Character : 글자와 문장 데이터 인식, 특수기호 포함

```
char1 <- "a" ; class(char1)
```

```
## [1] "character"
```

```
char2 <- "character" ; class(char2)
```

```
## [1] "character"
```

```
char3 <- "3" ; class(char3)
```

```
## [1] "character"
```

```
char4 <- "year: 2020" ; class(char4)
```

```
## [1] "character"
```

1.4. Logical : 참, 거짓(True or False)의 논리판단

```
# 논리형 데이터(Logical Data)는 참/거짓의 두가지만 존재함  
logic1 <- TRUE ; class(logic1)
```

```
## [1] "logical"
```

```
logic2 <- T ; class(logic2)
```

```
## [1] "logical"
```

```
logic3 <- FALSE ; class(logic3)
```

```
## [1] "logical"
```

```
logic4 <- F ; class(logic4)
```

```
## [1] "logical"
```

```
logic5 <- 4>5 ; logic5 ; class(logic5)
```

```
## [1] FALSE
```

```
## [1] "logical"
```

```
logic6 <- 7>2 ; logic6 ; class(logic6)
```

```
## [1] TRUE
```

```
## [1] "logical"
```

1.5. Special Value

```
NA # NA : 결측값, 데이터가 없는 경우
```

```
## [1] NA
```

```
NaN # NaN : 불가능한 값(e.g. , 10/0)
```

```
## [1] NaN
```

```
-Inf * 3 # +/- Inf : +/-로 무한대 값
```

```
## [1] -Inf
```

1.6. numeric data 와 complex data의 연산

```
comp + num
```

```
## [1] 5+3i
```

```
comp - num
```

```
## [1] -1+3i
```

```
comp * num
```

```
## [1] 6+9i
```

```
comp / num
```

```
## [1] 0.666667+1i
```

```
comp * 1i
```

```
## [1] -3+2i
```

```
log(comp)
```

```
## [1] 1.282475+0.982794i
```

```
sqrt(comp)
```

```
## [1] 1.674149+0.895977i
```

2. Vector

2.1. vector 간의 연산

```
vec1 <- c(2, 4, 1, 3, 4, 5, 1, 2, 3, 5)
vec2 <- c(4, 5, 2, 3, 8, 3, 4, 1, 5, 2)

vec1 + vec2
```

```
## [1] 6 9 3 6 12 8 5 3 8 7
```

```
vec1 - vec2
```

```
## [1] -2 -1 -1 0 -4 2 -3 1 -2 3
```

```
vec1 * vec2
```

```
## [1] 8 20 2 9 32 15 4 2 15 10
```

```
vec1 / vec2
```

```
## [1] 0.500000 0.800000 0.500000 1.000000 0.500000 1.666667 0.250000 2.000000
## [9] 0.600000 2.500000
```

```
vec1 > vec2
```

```
## [1] FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE
```

```
vec1 >= vec2
```

```
## [1] FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE
```

```
12 + vec1
```

```
## [1] 14 16 13 15 16 17 13 14 15 17
```

```
12 / vec1
```

```
## [1] 6.0 3.0 12.0 4.0 3.0 2.4 12.0 6.0 4.0 2.4
```

2.2. character vector

```
# 문자 및 문장으로 이루어진 데이터 종류 (특수문자 포함)  
# Vector 에 문자와 숫자가 함께 입력되면 숫자도 문자로 취급  
# "" (쌍따옴표)로 데이터 입력
```

```
char_vec1 <- c("a", "b", "c") ; class(char_vec1)
```

```
## [1] "character"
```

```
char_vec2 <- c("year", 2020); class(char_vec2)
```

```
## [1] "character"
```

2.3. logical vector

```
logic_vec1 <- 1:9 > 5 ; logic_vec1 ; class(logic_vec1)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
```

```
## [1] "logical"
```

```
logic_vec2 <- c(T,F,F,T,F,T,F,T,T,F) ; logic_vec2 ; class(logic_vec2)
```

```
## [1]  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE
```

```
## [1] "logical"
```

```
# logical data의 연산  
# T or TRUE 는 1로 계산  
# F or FALSE 는 0으로 계산
```

```
T + T
```

```
## [1] 2
```

```
TRUE * FALSE
```

```
## [1] 0
```

```
sum(T, T, F, T, F)
```

```
## [1] 3
```

2.4. vector의 생성: 수열

```
# 1 부터 9까지 1간격으로 증가하면 수열 생성
1:9
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

```
#1 부터 9까지 1간격으로 증가하며 수열 생성
seq(from = 1, to=9, by=1)
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

```
#1 부터 9까지 3간격으로 증가하며 수열 생성
seq(from = 1, to = 9, by = 3)
```

```
## [1] 1 4 7
```

```
#1 부터 9까지 3간격으로 증가하며 수열 생성
seq(1, 9, 3)
```

```
## [1] 1 4 7
```

```
#1 부터 9까지 3등분하는 수열 생성
seq(1, 9, length.out = 3)
```

```
## [1] 1 5 9
```

2.5. vector의 생성: 원소 반복

```
rep(c(1, 2, 3), each = 4) # c(1, 2, 3)을 각각 4번 반복하기
```

```
## [1] 1 1 1 1 2 2 2 2 3 3 3 3
```

```
rep(c(1, 2, 3), time = 4) # c(1, 2, 3)을 4회 반복하기
```

```
## [1] 1 2 3 1 2 3 1 2 3 1 2 3
```

```
rep(c(1, 2, 3), each=4, time=4) # c(1, 2, 3)을 각각 4번씩 4회 반복하기
```

```
## [1] 1 1 1 1 2 2 2 2 3 3 3 3 1 1 1 1 2 2 2 2 3 3 3 3 1 1 1 1 2 2 2 2 3 3 3 3 1 1
## [39] 1 1 2 2 2 2 3 3 3 3
```

2.6 벡터에 저장된 값 추출 및 수정

```
a = 1:9 ; a
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

```
a[1] #a 의 첫번째 데이터
```

```
## [1] 1
```

```
a[1:4] #a 의 첫번째부터 네번째까지 순차적으로 데이터 불러오기
```

```
## [1] 1 2 3 4
```

```
a[c(1, 2, 5)] #a의 1, 2, 5번째 데이터 불러오기
```

```
## [1] 1 2 5
```

```
a[c(-2:-4)] #a의 2번째부터 4번째까지 데이터를 제외한 나머지
```

```
## [1] 1 5 6 7 8 9
```

```
a[a > mean(a)] # a에서 a의 평균보다 큰 데이터만 불러오기
```

```
## [1] 6 7 8 9
```

```
a[a == mean(a)] # a에서 a의 평균과 같은 데이터만 불러오기
```

```
## [1] 5
```

```
a[a < mean(a)] = 1 ; a # a 에서 평균보다 작은 값 바꾸기
```

```
## [1] 1 1 1 1 5 6 7 8 9
```

```
append(a, 10) # a 에 10을 추가
```

```
## [1] 1 1 1 1 5 6 7 8 9 10
```

```
append(a, 10, 2) # a 에 10을 추가 하되 두번째자리 뒤에 추가
```

```
## [1] 1 1 10 1 1 5 6 7 8 9
```

```
sort(a, decreasing = T) # a 를 내림차순으로 정렬
```

```
## [1] 9 8 7 6 5 1 1 1 1
```

```
sort(a, decreasing = F) # a 를 오름차순으로 정렬
```

```
## [1] 1 1 1 1 5 6 7 8 9
```

```
a[order(a, decreasing = T)] # a 를 내림차순으로 정렬(벡터의 첨자를 정렬)
```

```
## [1] 9 8 7 6 5 1 1 1 1
```

```
a[order(a, decreasing = F)] # a 를 오름차순으로 정렬(벡터의 첨자를 정렬)
```

```
## [1] 1 1 1 1 5 6 7 8 9
```

2.7. 통계함수

```
a #변수
```

```
## [1] 1 1 1 1 5 6 7 8 9
```

```
mean(a) # 평균
```

```
## [1] 4.333333
```

```
var(a) # 분산
```

```
## [1] 11.25
```

```
sd(a) # 표준편차
```

```
## [1] 3.354102
```

```
summary(a) # 통계적 요약
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.000   1.000   5.000   4.333   7.000   9.000
```

```
max(a) # 최댓값
```



```
## [1] 9
```

```
min(a) # 최솟값
```

```
## [1] 1
```

2.8. 기타 벡터 다루기

```
object.size(a)
```

```
## 176 bytes
```

```
length(a) # a의 길이 세기
```

```
## [1] 9
```

```
nchar("alphago") #문자의 길이 세기
```

```
## [1] 7
```

```
length("alphago")
```

```
## [1] 1
```

```
letters[1:5] #문자열 만들기
```

```
## [1] "a" "b" "c" "d" "e"
```

```
names(a) = c("c1", "c2", "c3", "c4", "c5", "c5", "c6") ; a #원소에 이름붙이기, 이름 안붙은 원소는 <NA> 처리
```

```
##      c1      c2      c3      c4      c5      c5      c6 <NA> <NA>
##      1       1       1       1       5       6       7      8      9
```

PR3 연습문제

문제 1

```
url <- "https://www.ajou.ac.kr/kr/ajou/notice.do?mode=list&&articleLimit=10&article.offset="
portal <- paste0(url, seq(from=0, to=90, by=10))
portal
```

```
## [1] "https://www.ajou.ac.kr/kr/ajou/notice.do?mode=list&&articleLimit=10&article.offset=0"
## [2] "https://www.ajou.ac.kr/kr/ajou/notice.do?mode=list&&articleLimit=10&article.offset=10"
## [3] "https://www.ajou.ac.kr/kr/ajou/notice.do?mode=list&&articleLimit=10&article.offset=20"
## [4] "https://www.ajou.ac.kr/kr/ajou/notice.do?mode=list&&articleLimit=10&article.offset=30"
## [5] "https://www.ajou.ac.kr/kr/ajou/notice.do?mode=list&&articleLimit=10&article.offset=40"
## [6] "https://www.ajou.ac.kr/kr/ajou/notice.do?mode=list&&articleLimit=10&article.offset=50"
## [7] "https://www.ajou.ac.kr/kr/ajou/notice.do?mode=list&&articleLimit=10&article.offset=60"
## [8] "https://www.ajou.ac.kr/kr/ajou/notice.do?mode=list&&articleLimit=10&article.offset=70"
## [9] "https://www.ajou.ac.kr/kr/ajou/notice.do?mode=list&&articleLimit=10&article.offset=80"
## [10] "https://www.ajou.ac.kr/kr/ajou/notice.do?mode=list&&articleLimit=10&article.offset=90"
```

문제

```
player <- c("조영욱", "엄원상", "고영준", "정우영", "정호연", "백승호", "박진섭")
player_height <- c(178, 173, 168, 180, 180, 182, 184)
player_team <- c("김천 상무 FC", "울산 현대", "포항 스틸러스", "VfB 슈투트가르트", "광주 FC",
"전북 현대 모터스", "전북 현대 모터스")
player_age <- c(24, 24, 22, 23, 22, 26, 27)
```

문제 2

```
names(player) <- c(10, 11, 13, 7, 15, 8, 4)
player
```

```
##      10      11      13      7      15      8      4
## "조영욱" "엄원상" "고영준" "정우영" "정호연" "백승호" "박진섭"
```

문제 3

```
player[player_age>24]
```

```
##      8      4
## "백승호" "박진섭"
```

문제 4

```
player[player_team == "전북 현대 모터스"]
```

```
##      8      4
## "백승호" "박진섭"
```

문제 5

```
player[player_height>=180]
```

```
##           7           15           8           4
## "정우영" "정호연" "백승호" "박진섭"
```

문제 6

```
mean(player_age)
```

```
## [1] 24
```

문제 7

```
player[player_age == max(player_age)]
```

```
##           4
## "박진섭"
```