



Day19; 20220929

📅 날짜	@2022년 9월 29일
📅 유형	@2022년 9월 29일
🏷️ 태그	

GitHub - u8yes/R

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

<https://github.com/u8yes/R>

u8yes/R



1 Contributor 0 Issues 1 Star 0 Forks

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/05636e90-5aca-48df-9aeb-177bc9ce254f/chap02_Data_Structure.r

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/70af22a0-f2d3-4e29-82af-801c279ff68a/02._%EB%8D%B0%EC%9D%B4%ED%84%B0_%EC%9C%A0%ED%98%95%EA%B3%BC_%EA%B5%AC%EC%A1%B0.pdf

배열

자바는 배열은 반드시 **동일한** 자료형으로 저장할 수 있다.

자바스크립트는 var 이기 때문에 **여러** 자료형을 넣을 수 있었다.

자바스크립트는 정수, 실수, 문자형, 배열도 한꺼번에 담을 수 있다.

R도 자료형이 없다. 하지만!!!! 동일 데이터 타입, 곧 같은 자료형으로 담아야 한다.

Combine 결합하다

동사

1. (하나의 단일체를 만들도록) 결합하다[되다]

[V] Hydrogen and oxygen **combine** to form water.

수소와 산소가 결합하여 물을 생성한다.

문형 ~ (sth) (with sth) | ~ A and B (together)

2. 타동사 [VN]

(두 가지 이상의 자질·특징 등을) 갖추다[겸(비)하다]

The hotel **combines** comfort with convenience.

그 호텔은 안락함과 편리함을 겸비하고 있다.

문형 ~ A and/with B

3. 타동사 [VN]

(두 가지 이상의 일을) 병행하다

The trip will **combine** business with pleasure.

그 여행은 업무와 관광을 병행하게 될 것이다.

문형 ~ A and/with B

4. (공동 작업을 위해) 단합[결합]하다[시키다]

[V] They **combined** against a common enemy.

그들은 공동의 적에 대항해 단합했다.

행렬의 구조는 행과 열 구조.

3차원은 X에서 Y로 연속으로 올라가는 것

1차원 벡터

```
11
12 x <- c(1:20) # 콜론 : 범위
13 x
14
```

```
> x
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
> y <- 10:20
```

c()는 생략가능하다.

```
14
15 y <- 10:20
16 y
17
```

```
> y
[1] 10 11 12 13 14 15 16 17 18 19 20
>
```

R에서는 숫자의 자료형은 Numeric이다, Integer가 아니다.

처음부터 실수를 기본으로 잡는다.

```
8 ## - c() 함수 이용 벡터 객체 생성
9 x <- c(1, 2.5, 3, 4, 5) # combine 함수
10 x
11
```

```
> x
[1] 1.0 2.5 3.0 4.0 5.0
>
```

sequence()함수

```
18 ## - seq() 함수 이용 벡터 객체 생성
19 x <- seq(1, 10, 2) # sequence(시작, 종료, 증감)
20 x
21
```

```
> x
[1] 1 3 5 7 9
>
```

영어공부

기출 표현 sorry for the **inconvenience** 불편을 드려 죄송합니다

예 We apologize for any **inconvenience** caused by the ongoing renovations. 진행 중인 보수작업으로 인해 불편을 끼쳐 드려 죄송합니다.

R 수업 관련 데이터 셋 설명

데이터 셋	Severity_Counts 데이터 셋				
Severity_Counts는 RSADBE 패키지에서 제공하는 데이터 셋으로 다음과 같이 소프트웨어 발표 전과 후의 버그를 측정한 10개의 벡터 자료를 제공한다.					
Bugs.BR/AR	NT.BR/AR	Major.BR/AR	Critical.BR/AR	H.BR/AR	
단순한 버그	사소하지 않음	중대한 버그	결정적인 버그	시급한 버그	

Bugs.BR/AR 그동안 있었던 버그. (Before/After)

BR 출시되기 전, AR 출시되기 후 (줄어드는) Bugs.

```
> # 패키지에서 제공되는 벡터데이터 셋 보기
> Severity_Counts
      Bugs.BR      Bugs.AR      NT.Bugs.BR      NT.Bugs.AR      Major.BR
      11605         374         10119          17         1135
Major.AR Critical.BR Critical.AR H.Priority.BR H.Priority.AR
      35          432          10          459          3
# 벡터의 행 우선으로 행렬 객체 생성
```

기본(열 우선) 테이블 기준

```
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    3    5    7    9   11
[2,]    2    4    6    8   10    1
# 벡터의 행 우선으로 행렬 객체 생성
```

행 우선

```
12
13 # 벡터의 행 우선으로 행렬 객체 생성
14 m <- matrix(c(1:11), nrow = 2, byrow = 1) # 행 우선선
15 m
16
```

```
> m
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    2    3    4    5    6
[2,]    7    8    9   10   11    1
>
```

오늘 진도나간 곳.

```
# chap02_DataStructure
#####
# chapter02. 데이터의 유형과 구조
#####
```

```

# 1. Vector 자료 구조
## - c() 함수 이용 벡터 객체 생성
x <- c(1, 2.5, 3, 4, 5) # combine 함수
x

x <- c(1:20) # 콜론 : 범위
x

y <- 10:20
y

## - seq() 함수 이용 벡터 객체 생성
x <- seq(1, 10, 2) # sequence(시작, 종료, 증감)
x

## - rep() 함수 이용 벡터 객체 생성
help(rep) # ?rep
example(rep)
rep(1:3, 3) # replicate(대상, 반복수)
# 1 2 3 1 2 3 1 2 3
rep(1:3, each=3)
# 1 1 1 2 2 2 3 3 3

# union(), setdiff(), intersect() 함수 이용
x <- c(1, 3, 5, 7)
y <- c(3, 5)
x; y

union(x, y)      # 합집합(x+y) # 1 3 5 7
setdiff(x, y)    # 차집합(x-y) # 1 7
intersect(x, y)  # 교집합(x^y) # 3 5

# 숫자형, 문자형, 논리형 벡터 생성
v1 <- c(33, -5, 20:23, 12, -2:3)
v1
v2 <- c(33, -5, 20:23, 12, "4") # 데이터를 모두 문자형으로 변환.
v2

# 한 줄에 명령문 중복 사용
v1; mode(v1) # mode()는 자료형을 반환해줌.
v2; mode(v2)

# 벡터에 컬럼명 지정
age <- c(30, 35, 40)
age
names(age) <- c("홍길동", "이순자", "강감찬치킨")
age
#   홍길동   이순자   강감찬치킨
#    30     35     40
age <- NULL # age 변수 데이터 삭제
age

# 벡터 자료 참조하기
a <- c(1:50)
a[10] # index : 1부터 시작
a[c(10:45)] # 10 ~ 45 사이의 벡터 원소 출력
a[c(10, 20, 30, 40)]
a[10:(length(a)-5)] # 모든 함수가 올 수 있다. 그것을 예상하고 있어야 한다.

# 잘못된 벡터 첨자 사용 예
a[5, 10] # Error in a[5, 10] : incorrect number of dimensions

# c() 함수에서 콤마 사용 예
v1 <- c(33, -5, 20:23, 12, -2:3)
v1
v1[3:6] # 20 21 22 23
v1[c(4, 5:8, 9)]

# 음수 값으로 첨자 지정 예
v1
v1[-1] # 첫번째 인덱스를 제외시키고 실행시켜라 = 해당 위치의 원소를 제외한 값 출력
# -5 20 21 22 23 12 -2 -1 0 1 2 3
v1
v1[-c(2,4)]

# 패키지 설치와 메모리 로딩
install.packages("RSADBE") # 패키지(데이터) 설치
library(RSADBE)           # 패키지를 메모리에 로드

data("Severity_Counts") # RSADBE 패키지에서 제공되는 데이터 셋 가져오기.
str(Severity_Counts) # Structure(구조)

```

```

# 패키지에서 제공되는 벡터데이터 셋 보기
Severity_Counts

# 2. Matrix 자료 구조
args(matrix) # (data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)

# 벡터 이용 행렬 객체 생성
m <- matrix(c(1:5))
m # 5행 1열

# 벡터의 열 우선으로 행렬 객체 생성
?matrix
m <- matrix(c(1:10), nrow = 2) # 2행 5열
m

# 행과 열의 수가 일치하지 않는 경우 예
m <- matrix(c(1:11), nrow = 2)
m

# 벡터의 행 우선으로 행렬 객체 생성
m <- matrix(c(1:11), nrow = 2, byrow = T) # 행 우선
m

m <- matrix(c(1:10), byrow = T) # 주의 - 여전히 10행 1열
# (한글) nrow 또는 ncol 중 하나가 주어지지 않으면 데이터 길이와 다른 매개 변수에서 추론하려고 시도합니다. 둘 다 주어지지 않으면 한 열 행렬이 반환됩니다.
# (english) If one of nrow or ncol is not given, an attempt is made to infer it from the length of data and the other parameter. If ne
# 즉 byrow 속성은 적용되지 않음.
m

m <- matrix(c(1:10), ncol = 10) # 의도한 1행 10열
m

# 행 묶음으로 행렬 객체 생성
x1 <- c(5, 40, 50:52)
x2 <- c(30, 5, 6:8)
mr <- rbind(x1, x2)
mr

# 열 묶음으로 행렬 객체 생성
mc <- cbind(x1, x2)
mc

# 2행으로 행렬 객체 생성
m3 <- matrix(10:19, 2)
m3

# 자료와 객체 type 보기
mode(m3) # "numeric"
class(m3) # "matrix" "array"

# 행렬 객체에 첨자로 접근
m3[2,3] # 2행 3열의 데이터 1개 : 15
m3[1,] # 1행 전체 # 10 12 14 16 18
m3[,5] # 18 19
m3[1, c(2:5)] # 1행에서 2-5열까지의 데이터 # 12 14 16 18
m3[1, c(2, 5)] # 1행에서 2열, 5열 데이터 2개 # 12 18

# 3행 3열로 행렬 객체 생성
x <- matrix(c(1:9), nrow = 3, ncol = 3) # nrow, ncol, nro, nco, nr, nc 모두 가능하다. 하지만 n, n은 안 된다.
x

# 자료의 개수 보기
length(x) # 데이터 개수
ncol(x); nrow(x) # 열 / 행 수

# apply() 함수 적용
apply(x, 1, max) # 1번은 행, 2번은 열 # 행 단위 최대값
apply(x, 1, min) # 행 단위 최소값
apply(x, 2, mean) # 열 단위 평균값

```


