

R 프로그래밍

(4주차)

2016. 03. 26(토)

장운호

(ADP 002-0004)

목차

※ 지난 주 복습

I . 벡터 인덱싱(Indexing)

II. 연산자(Operator)

III. 내장함수(Built-in Function)

※ 벡터 생성법

벡터는 R의 모든 데이터구조의 기본이 되는 형태로,
많은 연습을 통해 익숙하게 사용할 수 있도록 훈련해 둘 필요가 있음.

[벡터 생성 함수 소개 및 사용법]

벡터 생성 함수명	사용법
<code>“:”</code>	<code>from:end</code> # from부터 end까지의 정수를 벡터로 생성해줌
<code>seq</code>	<code>seq(from, to, by=1)</code> # from부터 to까지 by간격으로 벡터를 생성
<code>rep</code>	<code>rep(x, times, each=1)</code> # x를 times만큼 반복한 벡터를 생성
<code>sample</code>	<code>sample(x, y, replace=FALSE)</code> # x벡터에서 y번 반복하여 비복원(replace=FALSE) 또는 복원(replace=TRUE) 샘플링을 시행하여 벡터를 생성
<code>seq_along</code>	<code>seq_along(x)</code> #1부터 x의 원소의 개수까지 일련번호를 생성

※ 벡터화(Vectorization)

R에서 벡터를 사용하는 이유는 여러 개의 원소들을 한번에 처리하기 위함임.

- 이러한 기능이 구현된 함수들을 "벡터화(Vectorize)"되었다고 표현함.

```
1 + 2  
2 + 3  
3 + 4  
4 + 5  
5 + 6  
7 + 8  
9 + 10  
10 + 11  
11 + 12  
12 + 13  
13 + 14  
14 + 15  
15 + 16  
16 + 17
```



Vectorization

```
X <- 1:16  
Y <- X + 1  
X + Y
```



I. 벡터 Indexing

1. 벡터 Indexing 개요

벡터의 원소들을 개별적으로, 또는 부분집합으로 다룰 필요가 있을 때, 객체명 옆에 대괄호("[")를 열고, 아래의 Rule에 따른 Index방법을 표기한 후, 대괄호("]")를 닫음으로써 지정이 가능함

[벡터 인덱싱(Indexing) Rule 요약]

- 1) 양의 정수가 사용되면, 해당 위치의 원소를 의미한다.
- 2) 빈칸으로 둔 경우는 모든 원소를 지정하는 것이 된다.
- 3) 음의 정수가 사용되면, 해당 위치의 원소가 제외한다는 의미다.
- 4) 조건식을 넣으면 조건식의 참인 원소가 선택된다.
- 5) 정수로 이뤄진 벡터를 넣으면, 해당 벡터의 위치에 있는 원소를 선택한다.

2. 벡터 Indexing 연습

R에서 벡터의 첫번째 원소의 index는 "1" 임.

- Java나 Python 등의 프로그래밍언어에서는 0으로 시작됨.

```
str_vector <- c("hello,", "me","?", "it", "looking", "is", "you", "for")
```

```
bool_vector <- c(TRUE, FALSE, TRUE, FALSE, FALSE, TRUE, TRUE, FALSE)
```

```
length(str_vector); str_vector[1]; str_vector[3]
```

```
sum(bool_vector)
```

```
str_vector[4 : length(str_vector)]
```

```
str_vector[4 : length(str_vector) - 1]; str_vector[4 : (length(str_vector) - 1)]
```

```
str_vector[bool_vector]
```

3. 벡터 Indexing 연습 – for loop

for 문은 특정한 명령어를 반복적으로 실행할 때, 유용하게 사용 가능함.

- 벡터 자체 혹은 벡터의 인덱스 번호 등을 반복의 기준으로 사용 가능함.

```
str_vector <- c("hello,", "me","?", "it", "looking", "is", "you", "for")
```


```
for (i in str_vector) {  
  print(i)  
}
```

```
word_arrange <- c(1,6,4,2,7,5,8,3)  
for (i in word_arrange) {  
  print(str_vector[i])  
}
```


4. 벡터 관련 내장함수 정리

객체 관리 및 파일관리 등에 활용되는 함수들에 대한 연습도 함께 수행 필요

구분	기능	함수
벡터 생성	원소들을 벡터로 결합	c(x, ...)
	정수로 된 일련번호순의 벡터생성	from:to
	일정한 간격으로 Sequence 생성	seq(from, to, by)
	특정한 숫자 혹은 벡터를 반복	rep(x, times, each)
	특정 모집단에서 랜덤 샘플링된 벡터 생성	sample(x, y, replace=FALSE) #replace 키워드로 복원/비복원 구분
	동일한 랜덤 샘플링이 되도록 seed 생성	set.seed(임의의 정수)
벡터 관리	벡터의 길이(원소의 개수) 산출	length(x)
	상/하위 6개 데이터 확인	head(x); tail(x)
	조건에 맞는 벡터의 인덱스만 추출	which(x 조건비교 y)
	x벡터 원소중 y벡터에 없는 원소 추출	setdiff(x, y) # x,y의 순서가 중요
	벡터간의 교집합 / 합집합	intersect(x, y); union(x, y)
	중복제거 기준 벡터간의 동일원소 여부 비교	setequal(x, y)
	원소의 개수만큼 일련번호 생성	seq_along(x)



II. 연산자 (Operator)

1. 연산자의 종류

연산자의 활용은 프로그래밍의 기본이 되는 사항으로,
이를 활용하여 명령어 표현식(Expression)을 간결하게 코딩하는 능력이 중요함.

※ 모두 Vectorization이 적용되어 있음

[산술 연산자]

연산자	설명
+	더하기
-	빼기
*	곱하기
/	나누기
^ or **	제곱
x %% y	나머지 (5%%2 는 1)
x %/% y	몫 (5%/%2 는 2)

[논리 연산자]

연산자	설명
<	작다
<=	작거나 같다
>	크다
>=	크거나 같다
==	같다
!=	같지 않다
!x	Not x
x y	x OR y
x & y	x AND y
isTRUE(x)	test if X is TRUE

2. 연산자 우선순위(Precedence)

괄호를 활용하여 우선순위를 적절히 조절하는 것이 중요함.

연산자	설명	적용순서	참조
()			
^	N제곱 계산	오른쪽 → 왼쪽	
-X, +X	음수/양수 기호	왼쪽 → 오른쪽	
:	정수 벡터 생성	from:to	
%%	나머지 계산	"	%in%, %*%
*, /	곱하기, 나누기 계산	"	
+, -	더하기, 빼기 계산	"	
<, >, <=, >=, ==, !=	비교 연산자 (TRUE, FALSE 반환)	"	
!	부정 조건 (상동)	"	
&	And 조건 (상동)	"	&&
	Or 조건 (상동)	"	
->	우측 할당	"	-> >
<-	좌측 할당	오른쪽 → 왼쪽	<<-
=	좌측 할당	오른쪽 → 왼쪽	

()

상위

하위



Ⅲ. 내장함수

1. 주요 내장함수 I

객체 관리 및 파일관리 등에 활용되는 함수들에 대한 연습도 함께 수행 필요

구분	기능	함수
패키지 관리	패키지 인스톨	install.packages("패키지명")
	패키지 메모리 로딩	library(패키지명)
	패키지를 메모리에서 삭제	detach("package:패키지명", character.only=TRUE)
객체 관리	메모리상의 객체 리스트 추출	ls(x)
	메모리상의 객체 삭제	rm(x) 또는 rm(list="객체명")
	디렉토리내 파일 리스트 추출	dir("파일PATH")
형변환	데이터 모드(mode)전환	as.~~~~~(x)
조건 비교	벡터화된 ifelse 함수	Ifelse(조건문, 참일때 실행값, 거짓일때 실행값)
	원소가 모두/하나라도 주어진 조건을 만족하는지 여부 검출	all(x 조건비교 y), any(x 조건비교 y)
특이 데이터	결측값 여부 검출	is.na(x)
	미확정값(NULL) 여부 검출	is.null(x)

2. 주요 내장함수 II

주요 내장함수의 용도, Argument의 개수 등을 반복적인 연습을 통해 숙달 필요

구분	기능	함수
수식	절대값	abs(x)
	제곱근	sqrt(x)
	N제곱, n제곱근	x^n , $x^{(1/n)}$
	올림/내림	ceiling(x), floor(x)
	지수함수값	exp(x)
	소수점 n자리 반올림	round(x, digits=n)
	자연로그, 상용로그	log(x), log10(x)
	숫자 ↔ 문자열 전환	as.numeric("x"), as.character(x)
문자열	문자열 글자수	nchar("x")
	문자열 일부선택	substr("x", 시작위치, 끝위치)
	단어 붙이기	paste("x", "y", sep=" ")
		paste0("x","y")

3. 수식 및 내장함수 사용 연습

Sigmoid 함수

$$Prob1 = \frac{e^3}{1 + e^3}$$

$$Prob2 = \frac{e^{\sqrt[5]{9}}}{1 + e^{\sqrt[5]{9}}}$$

```
x <- seq(-20,20,0.1)
length(x)
sigmoid <- exp(x)/(1+exp(x))
plot(x, sigmoid)
```

Prob2 = 0.8251801

End of Document.

감사합니다.