

제 3장. Basic R commands

3장에서는 기본적인 R 명령어에 대해 살펴보기로 한다.

3.1 벡터 (vectors)

R에서 벡터를 생성하기 위해서는 `c()` 함수를 사용한다. 예를 들어 (1,3,5,7)로 구성된 벡터를 생성하기 위해서는 다음과 같이 실행한다. 벡터의 연산은 같은 위치에 있는 스칼라들끼리 이루어지며 두 벡터의 길이가 다른 경우에는 짧은 길이를 갖는 벡터가 반복이 된다.

```
Console ~/
> x<-c(1,3,5,7)
> y<-c(2,4,6)
> z<-x+y
warning message:
In x + y : longer object length is not a multiple of shorter o
bject length
> z
[1] 3 7 11 9
```

벡터의 특정 위치에 있는 원소(스칼라)만을 사용하기 위해서는 `[]` 기호를 사용한다. 예를 들어 `x[2]`의 의미는 `x`라는 벡터의 두 번째 원소를 선택한다는 의미이다. 부분적으로 선택된 벡터(또는 스칼라)는 또 다른 오브젝트로 저장이 가능하다.

```
Console ~/
> x[1]
[1] 1
> x2<-x[2]
> x2
[1] 3
> x[c(1,3,4)]
[1] 1 5 7
> x[-2]
[1] 1 5 7
```

- ▶ 벡터 `x`의 첫 번째 원소를 출력
- ▶ 벡터 `x`의 두 번째 원소를 `x2`에 저장 후 출력
- ▶ 벡터 `x`의 (1,3,4)번째 원소만 출력
- ▶ 벡터 `x`의 두 번째 원소를 제외 후 출력

일정한 규칙이 있는 벡터는 `seq()` 함수를 통해서 생성이 가능하다. `seq()` 함수의 사용은 다음과 같다.

```
Console ~/
> seq(from=1, to=10, by=1)
[1] 1 2 3 4 5 6 7 8 9 10
> seq(from=1, to=15, length=10)
[1] 1.000000 2.555556 4.111111 5.666667 7.222222
[6] 8.777778 10.333333 11.888889 13.444444 15.000000
> seq(from=1, by=2, length=10)
[1] 1 3 5 7 9 11 13 15 17 19
```

- ▶ 1부터 10까지 1의 간격으로 벡터 생성
- ▶ 1부터 15까지 길이가 10인 벡터 생성
- ▶ 1부터 2의 간격으로 길이가 10인 벡터 생성

a부터 b까지의 정수 벡터는 a:b를 사용하여 간단히 생성할 수 있다. 동일한 값이 반복되는 벡터의 생성은 rep() 함수를 사용한다. rep() 함수는 스칼라 뿐 아니라 벡터를 반복하는 것도 가능하다.

```

>
> 3:10
[1] 3 4 5 6 7 8 9 10
>
> 10:3
[1] 10 9 8 7 6 5 4 3
>
> rep(3,5)
[1] 3 3 3 3 3
>
> rep(c(1,2,3),2)
[1] 1 2 3 1 2 3
>

```

- ▶ 3부터 10까지의 정수 벡터 생성
- ▶ 10부터 3까지의 정수 벡터 생성
- ▶ 숫자 3을 다섯 번 반복하여 벡터 생성
- ▶ 벡터 (1,2,3)을 두 번 반복하여 벡터 생성

3.2 행렬(Matrices)

행렬은 열(columns)과 행(rows)으로 이루어진 2차원 벡터이다. 행렬의 생성은 matrix() 함수를 이용한다. 길이가 동일한 벡터를 이용해서 행렬을 생성하기 위해서는 cbind() 또는 rbind()의 함수를 이용한다. (1장 참조)

벡터와 마찬가지로 행렬도 인덱스를 이용한 부분적 접근이 가능하다. 단, 행렬은 행과 열로 이루어져 있으므로 [row, column]의 형태로 인덱스를 지정하도록 한다.

```

> a1<-c(1,2,3,4)
> a2<-c(5,6,7,8)
> a3<-c(9,10,11,12)
>
> mat<-rbind(a1,a2,a3)
> mat
      [,1] [,2] [,3] [,4]
a1      1      2      3      4
a2      5      6      7      8
a3      9     10     11     12
>
> mat[1,3]
a1
3
> mat[3,2]
a3
10
> mat[1:3,3]
a1 a2 a3
3  7 11
> mat[,3]
a1 a2 a3
3  7 11

```

- ▶ 벡터 a1,a2,a3를 생성한 후 rbind()를 이용하여 행렬을 생성함
- ▶ 1행 3열의 원소를 출력
- ▶ 3행 2열의 원소를 출력
- ▶ 1~3행 전체에 대해 3열의 원소를 출력
- ▶ 인덱스를 생략하면 전체를 의미하게 됨

3.3 함수 (Functions)

R에는 다양한 함수들이 내장되어 있다. R에 내장된 함수는 수학함수, 문자함수, 통계분포함수, 통계함수, 데이터 생성 함수 등이 있다. 다음은 주로 사용되는 함수들의 이름과 기능이다.

함수	기능	예제
abs(x) sqrt(x) round(x, digit=n) log(x)	절대값 제곱근 소수점 아래 n번째 자리로 반올림 자연로그	<pre> > x<-2.53 > y<-3.45 > z<-8 > > abs(x) [1] 2.53 > sqrt(z) [1] 2.828427 > > round(x,digit=1) [1] -2.5 > round(y) [1] 3 > > log(z) [1] 2.079442 > round(log(z), digit=3) [1] 2.079 </pre>
choose(n, r) factorial(n)	조합 (combination) 순열 (factorial)	<pre> > choose(5,3) [1] 10 > > factorial(5) [1] 120 > </pre>
paste("a","b","c")	주어진 문자열 붙이기	<pre> > > paste("a","b","c") [1] "a b c" > > paste("z=", z, " and log(z)=",round(log(z),digit=2)) [1] "z= 8 and log(z)= 2.08" > </pre>

3.4 연산자

연산자는 스칼라, 벡터, 행렬, 데이터 오브젝트의 계산을 위한 명령어이다. 수학 연산자는 계산 결과를 나타내고 논리 연산자는 실행 결과로써 참(True) 또는 거짓(False)의 논리값을 보여준다.

함수	기능	예제
+, -, *, /	사칙연산	<pre> > x<-7; y<-2 > > x^y [1] 49 > > x%%y [1] 1 > > x%/%y [1] 3 </pre>
^	승수연산	
%% %/%	나머지 나눈 몫	
<, <=, >, >=	조건에서 양변의 크기 대소	<pre> > x<-7; y<-2 > > x>y; x<y [1] TRUE [1] FALSE > > (x%%y)==1 [1] TRUE > > (x%/%y)!=3 [1] FALSE </pre>
==	조건에서 양변이 동일	
!=	조건에서 양변이 다름	
 &	or, 두 조건중 하나가 참 and, 두 조건 모두 참	

3.5 제어문

3.5.1. For 문

문장을 반복 실행하는 경우에 사용한다. 지정된 변수가 주어진 값 만큼 변화하면서 문장을 반복 시행한다. For 문의 구성은 다음과 같다.

for (변수 in 구간) { 문장 }	주어진 변수가 지정된 구간에서 변화하면서 문장을 반복하여 실행함
-------------------------	-------------------------------------

<pre>> for (i in 1:3) { x<-i+1 ; print(x)} [1] 2 [1] 3 [1] 4 > > for (j in 5:1) { print(j) } [1] 5 [1] 4 [1] 3 [1] 2 [1] 1 ></pre>	<ul style="list-style-type: none">▶ 함수 print()는 괄호 안의 오브젝트를 출력▶ 주어진 i 변수가 1,2,3으로 순차적으로 증가하면서 { }안의 문장이 반복 실행되고 있다.▶ 5:1 은 (5,4,3,2,1)의 벡터를 만드는 명령어이다▶ j 변수가 5,4,3,2,1로 순차적으로 변화하면서 주어진 문장을 반복 실행하고 있다.
---	--

3.5.2 If-Else 문

주어진 조건을 판단하여 조건이 참인 경우에 문장을 실행한다. 조건이 거짓인 경우에는 else 문 뒤의 문장을 실행한다.

If (조건) { 문장1 } else { 문장2 }	괄호안의 조건이 참이면 문장1을 실행하고 그렇지 않으면 문장2를 실행함
--------------------------------	---

<pre>> if (3<5) { print("yes")} [1] "yes" > > if (3>5) { print("yes")} > > if (3>5) { print("yes")} else { print("no")} [1] "no" ></pre>	<ul style="list-style-type: none">▶ (3<5)의 실행 결과는 참이므로 "yes"를 출력▶ (3>5)의 결과는 거짓이므로 출력결과가 없음▶ (3>5)의 결과가 거짓이므로 else 뒤의 문장을 실행하여 "no"를 출력
<pre>> for (i in 1:10){ + if (i%%2==0) {print(i)}} [1] 2 [1] 4 [1] 6 [1] 8 [1] 10 ></pre>	<ul style="list-style-type: none">▶ 주어진 i 변수는 1부터 10까지 순차적으로 증가하며 if 문장을 실행▶ i 값을 2로 나누었을 때 나머지가 0인 경우에만 i를 출력하고 그렇지 않은 경우에는 실행하지 않음

3.6 예제

임의의 정수로 구성된 벡터를 생성하기 위해서는 `sample()` 함수를 사용할 수 있다.

<code>sample(x, size, replace=F)</code>	지정된 <code>x</code> 벡터에서 <code>size</code> 만큼의 크기(갯수)를 선택한다. 비복원 추출이 기본 설정이며 복원추출을 원하는 경우에는 <code>replace=T</code> 로 변경한다.
---	---

예제 1. 1부터 100 사이에서 40개의 정수를 임의로 선택하여 저장하자.

1-1) 예제 1에서 만들어진 벡터를 사용하여 5행 8열의 행렬을 생성하시오.

1-2) 예제 1에서 생성된 행렬에서 2행과 3행만을 추출하여 저장하시오.

1-3) 예제 1에서 생성된 행렬에서 (1,4,7,8)열을 추출하여 저장하시오.

1-4) 예제 1에서 생성된 행렬의 7번째 열의 평균과 분산을 구하시오.

예제 2. 1부터 20 사이에서 1개의 숫자를 임의로 선택하여 저장하자. 이 숫자가 10 이상이면 “P”를 출력하고 10 미만이면 “NP”를 출력하는 코드를 작성해보시오.

예제 3. 1부터 20 사이에서 8개의 숫자를 임의로 선택하여 저장하자. 8개의 숫자들 각각에 대해 숫자가 10 이상이면 “P”를 출력하고 10 미만이면 “NP”를 출력하는 코드를 작성해보시오.