초심자를 위한 생물학+정보학

_53_

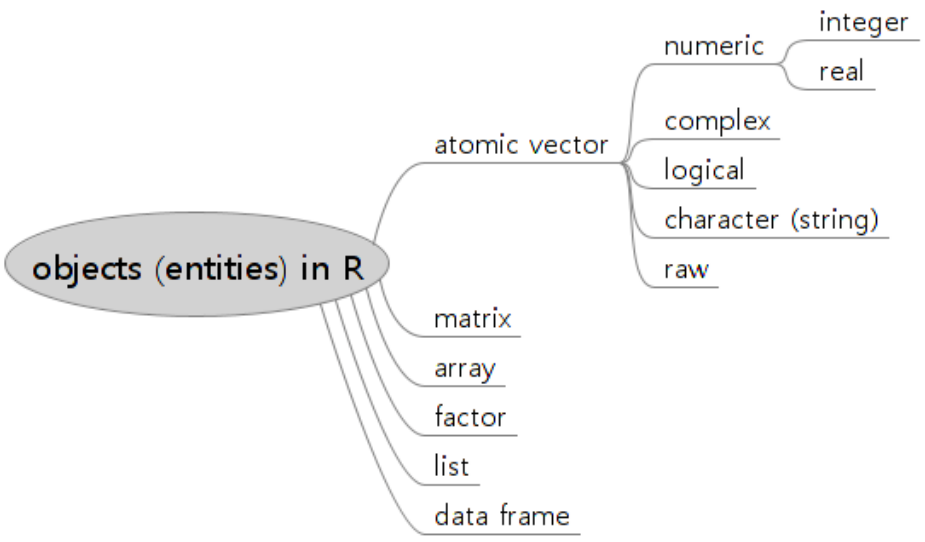
- R의 자료 형태와 구조 1 -

고 주 온 (John Go, Ph.D.)

지난 번에 통계학 도구인 R의 기동과 간단한 작동의 예를 보았다. 앞으로 다양한 도표에 관한 내용을 풀어 가면서 R을 주로 사용하게 될 것이므로 이 도구에 대하여 기초적인 사항을 알아보자.

기본적으로 R은 자료의 분석을 목적으로 설계된 관계로, 다른 범용 프로그래밍 언어보다는 다양한 자료 형태와 구조를 가지고 있다. 따라서 R의 도표 작성 기능을 사용함에 있어서도 R에서 사용하는 자료 구조에 대해서 어느 정도 알아야 할 필요가 있다. 그래서 지금부터 R의 자료 운용 방식에 대해서 잠시 알아 보고자 한다.

R에서 사용하는 자료 객체 (object)를 그림으로 요약 정리하면 다음과 같다.



R의 가장 기본적인 형식은 벡터 (vector)이다. 벡터의 종류에는 크게 수치형 (numeric), 복소수형 (complex), 논리형 (logical), 문자형 (문자 / 문자열, character / string), 그리고 메모리에 저장되어 있는 자료를 표현하는 raw 형 등이 있으며, 수치형은 다시 정수형 (integer)과 실수형 (real)으로 나눌 수 있다. 다만, 하나의 벡터를 이루는 원소는 모두 동일한 자료형이어야 한다. 예컨대, 수치와 문자열을 원소로 갖는 벡터는 만들 수 없다. 더 자세한 내용에 대해서는 나중에 다시 알아 볼 기회가 있을 것이다. 이 가운데 자주 접하게 되는 자료형은 주로 수치형 (numeric)과 문자형 (character)이다.

이외에도 행렬 (matrix), 배열 (array), 요인 (factor), 리스트 (list), 그리고 데이터프레임 (data frame) 등의 객체가 있다. 본 연재는 R의 다양한 기능 가운데 도표 작성을 중점적으로 다룰 예정이므로, 자료 객체에 대해서는 기본적인 형식인 벡터에 대해서 우선 알아 보고 나머지는 필요할 때마다 알아 보기로 하자.

R 에서 주어진 객체에 대하여 이들 자료형을 확인하는 함수로는 이미 앞 회차에서 소개한 mode()와 class() 외에도 typeof() 등이 있다. 이 함수들은 일반 사용자 입장에서 크게 다를 바가 없으나, 내용 면에서는 다음과 같은 차이점이 있다.

- mode(): Becker, Chambers & Wilks 등의 관점에서 본 각 객체의 자료형 (Becker et al., 1988).
- typeof(): 해당 객체의 저수준 (low-level) 자료형.
- class(): 객체 지향의 관점에서 본 고수준 (high-level) 자료형.

우선, 위의 여러 가지 자료형 중에서도 기본적 자료형인 벡터 (수치형, 복소수형, 논리형, 문자형, raw 형)에 대해서 알아 보자. 다음은 R 프롬프트 상태에서 벡터에 해당하는 각 자료형을 확인하는 내용이다. 각 함수의 차이를 이해하는 데에도 도움이 될 것이다.

```
------
> i <- 3  <-- (1)
> iL <- 3L  <-- (2)
> j <- 3.14  <-- (3)
> c <- 2 - 3i  <-- (4)
> b <- TRUE  <-- (5)
> s <- "09@AZaz"  <-- (6)
> r <- charToRaw(s)  <-- (7)
> mode(i)  <-- (8)
[1] "numeric"  <-- (9)
> typeof(i)  <-- (10)
[1] "double"  <-- (11)
> class(i)  <-- (12)
[1] "numeric"  <-- (13)
> mode(iL)  <-- (14)
[1] "numeric"  <-- (15)
> typeof(iL)  <-- (16)
[1] "integer"  <-- (17)
> class(iL)  <-- (18)
[1] "integer"  <-- (19)
> mode(j)  <-- (20)
[1] "numeric"  <-- (21)
> typeof(j)  <-- (22)
[1] "double"  <-- (23)
> class(j)  <-- (24)
[1] "numeric"  <-- (25)
> typeof(c)  <-- (26)
[1] "complex"  <-- (27)
> typeof(b)  <-- (28)
[1] "logical"  <-- (29)
> typeof(s)  <-- (30)
[1] "character"  <-- (31)
> typeof(r)  <-- (32)
[1] "raw"  <-- (33)
> print(s)  <-- (34)
[1] "09@AZaz"  <-- (35)
> print(r)  <-- (36)
[1] 30 39 40 41 5a 61 7a  <-- (37)
> length(i)  <-- (38)
```

```
[1] 1  <-- (39)
> length(s)  <-- (40)
[1] 1  <-- (41)
> length(r)  <-- (42)
[1] 7  <-- (43)
> s1 <- "Hello world!"  <-- (44)
> length(s1)  <-- (45)
[1] 1  <-- (46)
> nchar(s1)  <-- (47)
[1] 12  <-- (48)
> nchar(j)  <-- (49)
[1] 4  <-- (50)
> nchar(c)  <-- (51)
[1] 4  <-- (52)
>
------
```

변수 i에 3을 할당한다 (1). 이어서, 변수 iL[1]에 정수형 수치 3L 을 할당 (2), 변수 R에 실수 3.14 (3), 변수 c에 복소수 2-3i (4), 변수 b에 TRUE (5), 변수 s에 문자열 09@AZaz (6), 변수 r에는 앞에서 할당한 변수 s를 인수로 하여 함수 charToRaw()의 값 (7)을 각각 할당한다.

변수 i의 자료형을 알아 보기 위해서 함수 mode()를 적용하면 (8), 변수 i의 자료형이 수치형 (numeric)이라고 출력된다 (9). 동일한 변수 i에 대하여 자료형을 확인하는 저수준 자료형 함수인 typeof()를 적용하면 (10) 부동소수점 실수형인 double로 나타나는데 (11), 이에 대해서는 잠시 후에 실수형 변수와 함께 알아 보기로 하자. 계속하여 고수준 자료형 함수 class()를 사용하면 (12) 변수 i의 자료형이 수치형임을 알 수 있다 (13). 또 정수 3L이 할당된 변수 iL의 자료형을 mode()로 알아 보면 (14) 수치형이라고 나오지만 (15), typeof()나 class()를 적용하면 (16, 18) 정수형 (integer)이라고 출력된다 (17, 19).

함수 mode()로 실수 3.14가 들어 있는 변수 j의 자료형을 알아 보면 (20), 수치형임을 알 수 있다 (21). 그런데, 이러한 변수 j에 대하여 함수 typeof()와 class()로 그 자료형을 확인해 보면 (22, 24), 부동소수점 실수형 (23)과 수치형 (25)임을 알 수 있다. 이와 같이 R에서는 그 값이 실수 3.14인 변수 j뿐만 아니라, 숫자 3이 할당된 변수 i까지도 부동소수점 실수형으로 취급한다. 다만, 인덱스나 순위 등 명백하게 정수가 필요한 경우에는 예에서의 변수 iL과 같은 정수형 변수를 사용한다. 각 변수의 크기를 확인하고 싶다면 object.size() 함수를 사용하면 된다.

다음에는 복소수 2-3i가 할당되어 있는 변수 c의 자료형은 복소수 (complex)이며 (26, 27), TRUE가 그 값으로 할당되어 있는 변수 b의 자료형은 논리형 (logical) (28, 29), 문자열 (09@AZaz)이 들어 있는 변수 s의 자료형은 문자형 (character) (30, 31), 그리고 charToRaw() 함수를 이용하여 raw 형 자료를 할당받은 변수 r의 자료형이 raw 형임을 밝혀 주고 있다 (32, 33). 이는 print() 함수를 이용하여 변수 s에 할당된 내용을 출력하여 확인할 수 있고 (34, 35), 또한 변수 s를 문자형 자료에서 raw 형으로 변환하여 (7), 그 결과인 변수 r의 내용을 출력함으로써 (36, 37) 알아 볼 수 있다.

---

[1] 정수형 변수명에 L을 붙일 필요는 없다. 일반적인 변수명을 사용하면 된다. 본문의 예에서는 편의 상 변수의 구분을 쉽게 하기 위해 L을 붙인 것이다.

한편, 각 변수의 길이가 중요한 정보가 되기도 하는데, 이는 `length()` 함수를 이용하면 해결할 수 있다. 즉, 정수와 문자열 변수의 길이는 1이지만 (38-41), 문자열의 raw 형 자료를 할당받은 변수의 길이는 1이 아니다 (42, 43). 또, 공백을 포함하여 12자 길이의 문자열을 변수 `s1`에 할당하고 (44) 그 길이를 확인하면 1이 나오는데 (45, 46), 이때 문자열의 내부 문자 개수를 확인할 필요가 있다면 `nchar()` 함수를 사용하면 된다 (47, 48). 참고로, 실수 `j`와 복소수 `c`의 길이는 공백 문자를 제외한 글자 수이다 (49-52).

이상으로 `R`의 기본 자료형인 벡터에 대해서 간략하게 알아 보았으며, 다음 회차에는 남은 자료형 가운데 행렬 (matrix)과 리스트 (list)의 간단한 예를 살펴보자. 나머지 형식의 자료 객체들 가운데 지난 회차의 예에서 잠시 보았던 데이터프레임 (data frame)과 배열 (array), 요인 (factor) 등의 자료형에 관해서는 추후에 도표 관련 내용을 알아 보면서, 적절한 기회에 예시와 함께 자세하게 다루어 보기로 한다.

<References>

Becker,R.A. *et al.* (1988) The New S Language: A Programming Environment for Data Analysis and Graphics Wadsworth & Brooks/Cole Advanced Books & Software.

고주온(필명) (http://bioprofiler.tistory.com)
IBM-XT시절부터 개인용 컴퓨터를 사용하였으나, 강산이 변한 지금도 어제 코딩 한 내용을 오늘 기억하지 못하는 자유인. 박사학위는 분자유전학 분야로 받았으며, 물리학과 화학에 관심만 있음. 현재 대학 교수로 재직 중.

고주온 (2020). [초심자를 위한 생물학+정보학] _53_R 의 자료 형태와 구조 1
Available from https://www.ibric.org/myboard/read.php?Board=news&id=317443&SOURCE=6 (May.22, 2020)

Email: member@ibric.org