more_container_types

September 23, 2020

More Container Types in R
Paul Stey, Ph.D.
2020-09-15

1 1. Review

- Container types
  - `vector`
  - `matrix`
- Indexing
  - Integers
  - Ranges
  - Booleans

1.1 1.1 Indexing `vector` (review)

```
In [3]: v <- c(4, 2, 111)        # create vector with 3 integers

        v[2]                     # get second element
```

2

```
In [6]: v[3] <- 99999            # modify 3rd element in vector

        print(v)
```

```
[1]     4     2 99999
```

1.2 1.3 Indexing/Slicing `matrix` (review)

```
In [7]: m <- matrix(rnorm(6), nrow = 3)

        print(m)
```

```
           [,1]       [,2]
[1,] -0.1554313 -1.2069182
[2,]  1.3551700  0.4251780
[3,]  0.1822087 -0.4378254
```

```
In [9]: m[3, 2]
```

-0.437825352129149

1.2.1 1.3.1 Slicing `matrix` (cont.)

```
In [10]: print(m)
```

```
           [,1]       [,2]
[1,] -0.1554313 -1.2069182
[2,]  1.3551700  0.4251780
[3,]  0.1822087 -0.4378254
```

```
In [11]: m[, 2]                    # get second column
```

1. -1.2069182162559 2. 0.425177994858632 3. -0.437825352129149

```
In [13]: m[3, ] <- 999             # replace 3rd row elements with 999

         print(m)
```

```
            [,1]       [,2]
[1,]  -0.1554313  -1.206918
[2,]   1.3551700   0.425178
[3,] 999.0000000 999.000000
```

2 2. The `array` Type

- *n*-dimensional array
- The generalization of `vector` and `matrix`

```
In [14]: dat <- 1:24                        # vector integers from 1 to 24

         w <- array(dat, dim = c(3, 4, 2))
```

```
In [16]: print(w)
```

```
, , 1

     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
```

```
[2,]    2    5    8   11
[3,]    3    6    9   12

, , 2

     [,1] [,2] [,3] [,4]
[1,]   13   16   19   22
[2,]   14   17   20   23
[3,]   15   18   21   24
```

2.1 2.1 Indexing/Slicing `array`

```
In [18]: w[3, 1, 2]                # 3rd row, 1st column, 2nd "layer"
```

15

```
In [19]: print(w[, ,2])            # all of second layer
```

```
     [,1] [,2] [,3] [,4]
[1,]   13   16   19   22
[2,]   14   17   20   23
[3,]   15   18   21   24
```

2.1.1 2.1.1 Assignment of `array` Elements

```
In [20]: w[, , 2] <- 999           # set all elements in 2nd layer to 999

In [21]: print(w)

, , 1

     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

, , 2

     [,1] [,2] [,3] [,4]
[1,]  999  999  999  999
[2,]  999  999  999  999
[3,]  999  999  999  999
```

3 3. The `list` Type

- One-dimensional container
- Elements can be heterogeneous (i.e., all different types)
- Often used "under-the-hood" in R, probably because it's very versatile

```
In [26]: d <- list(4, c(4.1, 3.2, 1.4), "potato")

In [27]: print(d)

[[1]]
[1] 4

[[2]]
[1] 4.1 3.2 1.4

[[3]]
[1] "potato"
```

3.1 3.1 Accessing `list` Elements

- Use double square bracket notation (i.e., `[[ ]]`)
- Call also use named elements

```
In [25]: d[[2]]                    # get second list elements
```

1. 4.1 2. 3.2 3. 1.4

```
In [28]: d[[2]][3]                 # Get third element from list's 2nd element
```

1.4

3.1.1 3.1.1 Name Elements in `list`

```
In [29]: w <- list(ham = "on rye",
                   turkey = rnorm(4),
                   egg = d)

In [37]: w$egg[[3]]
```

'potato'