1-03 Objects, modes, and types

CSI 500

Course material derived from:
An Introduction to R. Notes on R: A Programming Environment for Data Analysis and Graphics
Version 3.4.3 (2017-11-30)
https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf

Mode and Length

- R uses the vector as its foundational data type
  - all R vectors must be the same type or "mode"
  - types may be numeric, complex, logical, character, raw
- R also supports recursive data structures of mode "list"
  - lists may contain any type including lists
  - functions may include functions
  - expressions may include expressions

```
# Modes and length examples
> x = c(1, 2.3, 4.5, 6.789)
> x
[1] 1.0 2.3 4.5 6.789
> mode(x)
[1] "numeric"
> length(x)
[1] 3
>
> y = c(1, "happy", sqrt(9))
> y
[1] "1"     "happy" "3"

> mode(y)
[1] "character"
> length(y)
[1] 3
>
```

Changing object length

- Objects can have a mode without any data
- length can be increased by adding values beyond the current index range
  - any missing intermediate indexes get NA
- length can be decreased by overwriting with fewer values than the current index range
  - unused index values are discarded

```
# object length example
> x = numeric()
> x
numeric(0)
> length(x)
[1] 0
> mode(x)
[1] "numeric"
>
> x[1] = 5
> x[2] = 7
> x[5] = 13
>
> x
[1]  5  7 NA NA 13
>
> x = 11:13
> x
[1] 11 12 13
>
```

Object class

- Entities in R have an associated "class"
    - supports object-oriented programming style
    - facilitates default operations
- Simple vectors, the class is the same as the mode
    - numeric, complex, logical, character, raw
- More complex data structures have other class types
    - matrix, array, factor, data.frame

```
# class example
> x = 1:8
> x
[1] 1 2 3 4 5 6 7 8
> class(x)
[1] "integer"
>
> y = 2.5 * x
> y
[1]  2.5  5.0  7.5 10.0 12.5 15.0 17.5 20.0
> class(y)
[1] "numeric"
>
> z = matrix(nrow=2, ncol=4, data=x)
> z
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
> class(z)
[1] "matrix"
>
```

Object attributes

- Entities in R have "attributes"
  - important elements used to manage the object itself
- Example: a vector can have NULL for attributes
- Example: a matrix has a $dim attribute which specifies its row and column dimensions

```
# attributes example
> x = 1:4
> x
[1] 1 2 3 4
> attributes(x)
NULL
>
> xmat = matrix(nrow=2, ncol=2, data=x)
> xmat
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> attributes(xmat)
$dim
[1] 2 2
```

Summary

- R is based on vectors
  - vectors are 1-D arrays of elements of the same fundamental type
  - numeric, character, logical
- R components are "objects", which are instances of a class
  - allows object-oriented features for programming and software design
  - objects have attributes based on their class type