Introduction to R Data Types and Structures

Presented at Saint Louis RUG
July 20, 2016

Fundamental Data Types

- Common:
  - Logical (32-bit signed integer)
  - Integer (32-bit signed integer)
  - Double (64-bit floating point)
  - Character
- Less common:
  - Complex (2 doubles: 1 real, 1 imaginary)
  - Raw (raw binary)
  - NULL

Fundamental Data Types

- R refers to these types as “atomic”
- Doubles are usually referred to as “numeric”
  - Beware!
    - `is.numeric(as.integer(1)) # TRUE`
    - `is.double(as.integer(1)) # FALSE`

Common Derived Data Types

- Date
  - Integer (generally, but can be numeric)
- Factor
  - Integer with a “levels” attribute
- POSIXct
  - Numeric with possible “tzone” attribute

Fundamental Data Structures

- vector
- matrix
- array
- list
- data.frame
- environment

V is for vector

- `c()` is for concatenate, or combine
  - Can only contain one data type
  - All inputs coerced to "highest type" of the components `NULL < raw < logical < integer < double < complex < character` `(< list < expression)`
- Only 1 "dimension": length
  - `dim(c(0, 1))  # NULL`
  - `length(c(0, 1))  # 2`
  - Use `NROW` and `NCOL` if you want to be safe

V is for vector

- Vector elements can be named:

```
R> c(one=1, two=2)
one two
  1   2
R> x <- c(1, 2)
R> setNames(x, c("one", "two"))
one two
  1   2
R> names(x) <- c("one", "two")
R> x
one two
  1   2
```

M is for matrix

- A matrix is a vector with a “dim” attribute

```
R> matrix(1:10, nrow=2, ncol=5)
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
R> matrix(1:10, nrow=2, ncol=5, byrow=TRUE)
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    6    7    8    9   10
```

- `dim`, `nrow`, `ncol`
- `length` returns total number of elements
- Does not require row names or column names

M is for matrix

- Does not require row names or column names
  - Use `colnames` and `rownames` to access
  - Use `colnames<-` and `rownames<-` to set
  - Use `dimnames` to access/set both column names and row names at the same time (as a list)

    ```
    R> x <- matrix(1:6, 3, 2)
    R> dimnames(x) <- list(letters[1:3], c("a", "b"))
       a  b
    a  1  4
    b  2  5
    c  3  6
    ```

A is for array

- An array is a structure with more than 1 dimension
  - A matrix is an array…
    but “array” usually means more than 2 dimensions

```
R> array(1:8, c(2,2,2))
, , 1

     [,1] [,2]
[1,]    1    3
[2,]    2    4

, , 2

     [,1] [,2]
[1,]    5    7
[2,]    6    8
```

L is for list

- A “vector” of pointers to vectors
  - `is.vector(list())  # TRUE`
- Each list element can be a different type

```
R> list(1:5, head(letters), Sys.Date())
[[1]]
[1] 1 2 3 4 5

[[2]]
[1] "a" "b" "c" "d" "e" "f"

[[3]]
[1] "2016-07-20"
```

L is for list

- List elements can be almost anything

```
R> list(model=lm(1:10~rnorm(10)), thing=list(hi=1,there=2))
$model

Call:
lm(formula = 1:10 ~ rnorm(10))

Coefficients:
(Intercept)    rnorm(10)
     5.4602       0.9459

$thing
$thing$hi
[1] 1

$thing$there
[1] 2
```

D is for data.frame

- A data.frame is a list with specific components
  - Must have row names and column names
  - All columns must have the same number of rows
- Since they’re a list, you can have different types in each column of a data.frame
- Obligatory: `stringsAsFactors = FALSE`

E is for environment

- An environment is a set of name/value pairs
- The global workspace you’re used to working in is an environment
- Pass-by-reference semantics

Q is for questions