R DataTypes and Operations

author: Rohit Jain date: autosize: true

R atomic class

In R every thing is a object and for these object we have below atomic classes: - Character - Numeric - Integer - Complex - Logical

R DataTypes

Following are the data types in R:

- Vector
- List
- Matrices
- Data Frame
- Factors

Attributes of R objects

==========

Following are the attributes of R objects:

- names and dimnames
- dimensions
- class
- length
- other user defined attributes

attribute() function can be used to get the attribute of any R object.

Important functions

- getwd()
- setwd()
- objects()
- ls()
- rm()
- c()
- as.* functions (coercion)
- str()
- summary() etc....

Operation 1: creating Vector

```
x <- c(0.5,0.6)     ## numeric
x <- c("a","b","c") ## character
x <- c(TRUE,FALSE)  ## logical
```

```
x <- 1:30            ## integer
x <- c(1+2i,34-24i) ## complex
```

Using vector function

```
x <- vector("numeric", length = 10)
x
```

```
 [1] 0 0 0 0 0 0 0 0 0 0
```

Operation2: Mixing objects

```
x <- c(0.5,"a"); class(x)    ## character
```

```
[1] "character"
```

```
x <- c("a",FALSE); class(x)  ## character
```

```
[1] "character"
```

```
x <- c(TRUE,2); class(x)    ## numeric
```

```
[1] "numeric"
```

Operation3: Explicit coercion

Objects can be coerced explicitly using the as.* function from one class to another

```
x <- 0:6; class(x)
```

```
[1] "integer"
```

```
as.numeric(x)
```

```
[1] 0 1 2 3 4 5 6
```

```
as.logical(x)
```

```
[1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

Lists

```
l <- list("a",c(1,2,3,4),TRUE)
l
```

```
[[1]]
[1] "a"

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

[[3]]
[1] TRUE
```

Matrices1

```
m <- matrix(nrow = 2,ncol = 3); m
```

```
     [,1] [,2] [,3]
[1,]   NA   NA   NA
[2,]   NA   NA   NA
```

```
dim(m)
```

```
[1] 2 3
```

Matrices2

matrices are constructed column-wise

```
attributes(m)
```

```
$dim
[1] 2 3
```

```
m <- matrix(data = 1:6, nrow = 2, ncol = 3); m
```

```
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Matrices from vector

Matrices can be constructed from a vector using the dim() function

```
m <- 1:10; m
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

```
dim(m) <- c(2,5); m
```

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

cbind-ing and rbind-ing

```
x <- 1:3
y <- 10:12
cbind(x,y)
```

```
     x  y
[1,] 1 10
[2,] 2 11
[3,] 3 12
```

```
rbind(x,y)
```

```
  [,1] [,2] [,3]
x    1    2    3
y   10   11   12
```

Factors

Factors are used to refered to as categorical variables. Factors can be ordered or unordered. Factors in R are stored as a vector of integer values with a corresponding set of character values to use when the factor is displayed.

```
x <- factor(c("yes","no","yes","no","no","yes")); x
```

```
[1] yes no  yes no  no  yes
Levels: no yes
```

```
table(x)
```

```
x
 no yes
  3   3
```

Missing Values

Missing values in R are denoted by either NA or NAN - is.na is used to test objects are NA. - is.nan is used to test for NANs.

```
x <- c(1,2,3,NA); is.na(x)
```

```
[1] FALSE FALSE FALSE  TRUE
```

similarly we can check for Nans.

Data Frames

- Data frame is a two dimensional data structure in R. - It is a special case of a list which has each component of equal length.

```
x <- data.frame("SN" = 1:2, "Age" = c(21,15), "Name" = c("John","Dora")); str(x)
```

```
'data.frame':   2 obs. of  3 variables:
 $ SN  : int  1 2
 $ Age : num  21 15
 $ Name: Factor w/ 2 levels "Dora","John": 2 1
```

Names Attribute

```
x <- 1:3; names(x)
```

```
NULL
```

```r
names(x) <- c("a","b","c"); x
```

```
a b c
1 2 3
```

```r
names(x)
```

```
[1] "a" "b" "c"
```

========== lists can also be named

```r
l <- list("a" = 1 , "b" = 2, "c" = 3); l
```

```
$a
[1] 1

$b
[1] 2

$c
[1] 3
```

========== And matrices

```r
m <- matrix(data = 1:4,nrow = 2,ncol = 2)
dimnames(m) <- list(c("m","n"),c("a","b")); m
```

```
  a b
m 1 3
n 2 4
```

Reading and Writing Data

- read.table(), read.csv() reading tabular data - readLines() reads a line from the terminal - source() reads a R file - write.table() to write data in tabular format. - writeLines() Write text lines to a connection.

Connection to the outside world

- file, opens a connection to a file. - gzfile, opens a connection to a file compressed with gzip. - bzfile, opens a connection to a file compressed with bzip2. - url, opens a connection to a webpage

==========

```r
str(url)
```

```
function (description, open = "", blocking = TRUE, encoding = getOption("encoding"),
    method = getOption("url.method", "default"))
```

```r
con <- url("https://www.wikipedia.org/", "r")
x <-readLines(con)
head(x)
```

```
[1] "<!DOCTYPE html>"
[2] "<html lang=\"mul\" class=\"no-js\">"
[3] "<head>"
[4] "<meta charset=\"utf-8\">"
```

```
[5] "<title>Wikipedia</title>"
[6] "<meta name=\"description\" content=\"Wikipedia is a free online encyclopedia, created and edited by
```

Subsetting

We can use either [, [[or $ operator to access columns of data frame.

```
x <- data.frame("SN" = 1:2, "Age" = c(21,15), "Name" = c("John", "Dora"), stringsAsFactors = FALSE); x[
```

```
  Name
1 John
2 Dora
```

```
[1] "John" "Dora"
```

```
[1] "John" "Dora"
```

Removing Missing values

```
x <- c(1,3,NA,5,NA); bad <- is.na(x); x[!bad]
```

```
[1] 1 3 5
```

```
y <- c("a","b",NA,"m","n"); good <- complete.cases(x,y); x[good] ; y[good]
```

```
[1] 1 3 5
```

```
[1] "a" "b" "m"
```

==========

```
airquality[4:6,]
```

```
  Ozone Solar.R Wind Temp Month Day
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
```

```
good <- complete.cases(airquality)
airquality[good,][4:6,]
```

```
  Ozone Solar.R Wind Temp Month Day
4    18     313 11.5   62     5   4
7    23     299  8.6   65     5   7
8    19      99 13.8   59     5   8
```

Vectorized operations

```
x <- 1:4; y<-6:9
x+y; x>2 ; y == 8 ; x*y ; x/y
```

```
[1]  7  9 11 13
```

```
[1] FALSE FALSE  TRUE  TRUE
```

```
[1] FALSE FALSE  TRUE FALSE
```

```
[1]  6 14 24 36
```

```
[1] 0.1666667 0.2857143 0.3750000 0.4444444
```

Vectorized marix operations

```
x <- matrix(1:4,2,2); y <- matrix(rep(10,4),2,2)
x*y; x%*%y
```

```
     [,1] [,2]
[1,]   10   30
[2,]   20   40
```

```
     [,1] [,2]
[1,]   40   40
[2,]   60   60
```