Ch1. Playing with vectors

Jongrak Jeong

2023-01-04

Default Setting

```
setwd("~/Library/Mobile Documents/com~apple~CloudDocs/Study/2_Data Science/Practice/R Programming by He
```

1. Playing with vectors

Define the vectors

```
# make the vectors and put some elements
era <- c(5, 4, 3, 4, 5, 6) # numeric vectors (real numbers)
year <- 2001:2006 # integer vecotrs
game <- c(1, 2, 8, 6, 4, 4)

# print the vectors
era
```

```
## [1] 5 4 3 4 5 6
```

```
year
```

```
## [1] 2001 2002 2003 2004 2005 2006
```

```
game
```

```
## [1] 1 2 8 6 4 4
```

Basic operators of vectors

```
# The summation of game for 6 years
sum(game)
```

```
## [1] 25
```

```
# The average ERA for 6 years
era.all <- sum(game * era) / sum(game)
era.all
```

```
## [1] 4.2
```

```
# check whether it is a vector or not
is.vector(era)
```

```
## [1] TRUE
```

Integer and floating numbers

```
# check the class
class(era) # numeric is 'real number' or 'floating number'
```

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

```
class(game)
```

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

```
class(year) # integer is 'integer'
```

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

Subvector - indexing

```
# The second element of era vector
era[2]
```

```
## [1] 4
```

```
# The second and third elements of era vectors
era[2:3]
```

```
## [1] 4 3
```

```
# The entire elements of era vectors = era vectors
era[1:6]
```

```
## [1] 5 4 3 4 5 6
```

```
# The subvector is also vector
is.vector(era[2:3])
```

```
## [1] TRUE
```

Arithmetic operation

```
# numeric vectors
A <- c(2, 3, 4, 5, 6)
B <- c(1, 3, 5, 7, 9)

A + B
```

```
## [1]  3  6  9 12 15
```

```
A - B
```

```
## [1]  1  0 -1 -2 -3
```

```
A * B
```

```
## [1]  2  9 20 35 54
```

```
A / B
```

```
## [1] 2.0000000 1.0000000 0.8000000 0.7142857 0.6666667
```

```
class(A + B)
```

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

```r
class(A - B)
```

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

```r
class(A * B)
```

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

```r
class(A / B)
```

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

```r
# integer vectors
C <- 1:5
D <- 7:11

C + D
```

```
## [1]  8 10 12 14 16
```

```r
C - D
```

```
## [1] -6 -6 -6 -6 -6
```

```r
C * D
```

```
## [1]  7 16 27 40 55
```

```r
C / D
```

```
## [1] 0.1428571 0.2500000 0.3333333 0.4000000 0.4545455
```

```r
class(C + D)
```

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

```r
class(C * D)
```

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

```r
class(C / D) # integer / integer is numeric(real number)
```

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

```r
# numeric and integer vectors
A + C
```

```
## [1]  3  5  7  9 11
```

```r
class(A + C) # integer + numeric is numeric
```

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

2. Functions

Built-in function

```r
# sum() is a function
sum(game)
```

```
## [1] 25
```

```
length(game)
```

```
## [1] 6
```

User-defined functions

```
sum.1 <- function(x) {
  temp <- 0
  for (i in 1:length(x)) temp <- temp + x[i]
  return(temp)
}

sum.1(game)
```

```
## [1] 25
```

Application: counting the odd numbers in specific vecor x

```
# Basic version
oddcount1 <- function(x) {
  count <- 0
  for (i in 1:length(x)) {
    if (x[i] %% 2 == 1) count <- count + 1
  }
  return(count)
}

oddcount1(era)
```

```
## [1] 3
```

```
# 'count' is local variable

# Simple version
oddcount2 <- function(x) {
  return(sum((x %% 2) == 1))
}

oddcount2(era)
```

```
## [1] 3
```

3. Data containers

Vecotrs: contains only one kind of variables

```
# number: numeric and integer
era
```

```
## [1] 5 4 3 4 5 6
```

```
class(era)
```

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

```
# character strings
e <- c("Park", "LA Dodgers")
class(e)
```

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

```
# It's possible to combine some numbers and character strings. But it will become charcter strings vect
ex1 <- c(5, 4, 3, "LA Dodgers")

ex1
```

```
## [1] "5"          "4"          "3"          "LA Dodgers"
```

```
class(ex1)
```

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

Matrix

cbind: combine vectors - column arranging

```
# length: The number of rows
# the number of vectors: The number of columns

# Matrix with numeric variables
stat <- cbind(year, game, era)

class(stat)
```

```
## [1] "matrix" "array"
```

```
dim(stat)
```

```
## [1] 6 3
```

```
nrow(stat)
```

```
## [1] 6
```

```
ncol(stat)
```

```
## [1] 3
```

rbind: combine vectors - row arranging

```
# length: The number of columns
# the number of vectors: The number of rows

# Matrix with character variables
e <- c("Park", "LA Dodgers")
f <- c("Choo", "Cleveland Indians")
g <- c("Kang", "Pittsburgh Pirates", "?") # "? will be ommited in that it is out of the range (not same

M <- rbind(e, f, g)
```

```
## Warning in rbind(e, f, g): number of columns of result is not a multiple of
## vector length (arg 1)
```

```
M
```

```
##   [,1]   [,2]                 [,3]
## e "Park" "LA Dodgers"         "Park"
## f "Choo" "Cleveland Indians"  "Choo"
## g "Kang" "Pittsburgh Pirates" "?"
```

```
class(M)
```

```
## [1] "matrix" "array"
```

```
dim(M)
```

```
## [1] 3 3
```

```
nrow(M)
```

```
## [1] 3
```

```
ncol(M)
```

```
## [1] 3
```

List: the length and class of each vector is independent

```
# defining the list (without the names)
L <- list(game, era, e)

class(L)
```

```
## [1] "list"
```

```
# indexing the list "[["
L[[1]]
```

```
## [1] 1 2 8 6 4 4
```

```
"[["(L, 1)
```

```
## [1] 1 2 8 6 4 4
```

```
# defining the list with the names
L.1 <- list(Game = game, ERA = era, Player = e)
L.1
```

```
## $Game
## [1] 1 2 8 6 4 4
##
## $ERA
## [1] 5 4 3 4 5 6
##
## $Player
## [1] "Park"       "LA Dodgers"
```

```
# indexing the list with the names
L.1[[1]]
```

```
## [1] 1 2 8 6 4 4
```

```
L.1[["Game"]]
```

```
## [1] 1 2 8 6 4 4
```

```
L.1$Game
```

```
## [1] 1 2 8 6 4 4
```

```
names(L.1)
```

```
## [1] "Game"   "ERA"    "Player"
```

Data frame: The class of vectors can be different but the length should be same (a kind of list)

defining the data frame

```
year <- 2011:2014
winner <- c("SLN", "SFN", "BOS", "SFN")
loser <- c("TEX", "DET", "SLN", "KCA")
wins <- c(4, 4, 4, 4)
losses <- c(3, 0, 3, 3)

WS <- data.frame(year, winner, loser, wins, losses)
WS
```

```
##   year winner loser wins losses
## 1 2011    SLN   TEX    4      3
## 2 2012    SFN   DET    4      0
## 3 2013    BOS   SLN    4      3
## 4 2014    SFN   KCA    4      3
```

column indexing - variables

```
WS$winner
```

```
## [1] "SLN" "SFN" "BOS" "SFN"
```

```
WS[,2]
```

```
## [1] "SLN" "SFN" "BOS" "SFN"
```

```
WS[[2]]
```

```
## [1] "SLN" "SFN" "BOS" "SFN"
```

```
"[["(WS, 2)
```

```
## [1] "SLN" "SFN" "BOS" "SFN"
```

```
WS[2]
```

```
##   winner
## 1    SLN
## 2    SFN
## 3    BOS
## 4    SFN
```

```
"["(WS, 2)
```

```
##   winner
## 1    SLN
## 2    SFN
## 3    BOS
## 4    SFN
```

row indexing - objects

```
WS[3, ]
```

```
##   year winner loser wins losses
## 3 2013    BOS   SLN    4      3
```

```
WS["3", ]
```

```
##   year winner loser wins losses
## 3 2013    BOS   SLN    4      3
```

get info of data frame: str()

```
str(WS)
```

```
## 'data.frame':    4 obs. of  5 variables:
##  $ year  : int  2011 2012 2013 2014
##  $ winner: chr  "SLN" "SFN" "BOS" "SFN"
##  $ loser : chr  "TEX" "DET" "SLN" "KCA"
##  $ wins  : num  4 4 4 4
##  $ losses: num  3 0 3 3
```