L1

Sim

2018년 4월 15일

## Module 1 - Hello World

Hello World

## Error: unexpected symbol in "Hello World"

br

"Hello World"

## [1] "Hello World"

br

greeting <- "Hello World"  
greeting

## [1] "Hello World"

br

greeting + "Sir"

## Error in greeting + "Sir"

## : non-numeric argument to binary operator

br

paste(greeting, "Sir")

## [1] "Hello World Sir"

paste(greeting, "!")

## [1] "Hello World !"

paste0(greeting, "!")

## [1] "Hello World!"

paste(greeting, "!", sep="")

## [1] "Hello World!"

br

paste1(greeting, "sir")

## Error in paste1(greeting, "sir")

## : could not find function "paste1"

br

## Data Type - 1. string (character)

greeting <- "R says \"Hello World!\""  
nchar(greeting)

## [1] 21

substr(greeting,3,6)

## [1] "says"

greeting

## [1] "R says \"Hello World!\""

cat(greeting)

## R says "Hello World!"

br

paste(string1,string2)  
paste0(string1,string2,sep)  
nchar(string)  
substr(string,start,end)  
cat(string)

## Data Type - 2. numeric

10^2 + 36

## [1] 136

a = 4  
a

## [1] 4

a\*5

## [1] 20

a = a + 10  
a

## [1] 14

## Data Type - 3. logical

2==3

## [1] FALSE

5>3

## [1] TRUE

## Data Type - Type checking and conversion

is.character(5)

## [1] FALSE

is.character("5")

## [1] TRUE

a <- as.character(5)  
is.character(a)

## [1] TRUE

b <- as.numeric(a)  
is.numeric(b)

## [1] TRUE

as.numeric(2==3)

## [1] 0

## Data Type - Summary

## Data Structure - 1. vector (1/3)

strVec1 <- c("Hello", "Hi", "What's up")  
cbind(strVec1)

## strVec1   
## [1,] "Hello"   
## [2,] "Hi"   
## [3,] "What's up"

strVec2 <- c("Ma'am", "Sir", "Your Honor")  
cbind(strVec2)

## strVec2   
## [1,] "Ma'am"   
## [2,] "Sir"   
## [3,] "Your Honor"

strVec3 <- paste(strVec1, strVec2)  
cbind(strVec3)

## strVec3   
## [1,] "Hello Ma'am"   
## [2,] "Hi Sir"   
## [3,] "What's up Your Honor"

## Data Structure - 1. vector (2/3)

numVec1 <- c(30,50,70)  
numVec1

## [1] 30 50 70

numVec2 <- seq(30,70,20)  
numVec2

## [1] 30 50 70

numVec3 <- c(25,55,80)  
numVec3

## [1] 25 55 80

numVec4 <- seq(from=20, to=1, by=-3)  
numVec4

## [1] 20 17 14 11 8 5 2

2:6

## [1] 2 3 4 5 6

## Data Structure - 1. vector (3/3)

min(numVec1)

## [1] 30

min(numVec1,numVec3)

## [1] 25

pmin(numVec1,numVec3)

## [1] 25 50 70

pmin(numVec1 > numVec3)

## [1] TRUE FALSE FALSE

numVec1[2]

## [1] 50

numVec1[-2]

## [1] 30 70

numVec1[1:2]

## [1] 30 50

numVec1[c(1,3)]

## [1] 30 70

## Data Structure - 2. matrix

mat=matrix(data=c(9,2,3,4,5,6),ncol=3)  
mat

## [,1] [,2] [,3]  
## [1,] 9 3 5  
## [2,] 2 4 6

mat[1,2]

## [1] 3

mat[2,]

## [1] 2 4 6

mean(mat)

## [1] 4.833333

apply(mat,2,mean)

## [1] 5.5 3.5 5.5

apply(mat,1,mean)

## [1] 5.666667 4.000000

## Data Structure - 3. data.frame (1/2)

weather <-  
 data.frame(date = c("2017-8-31","2017-9-1","2017-9-2"),  
 sky = c("Sunny", "Cloudy", "Rainy"),   
 temp = c(20,15,18))  
weather

## date sky temp  
## 1 2017-8-31 Sunny 20  
## 2 2017-9-1 Cloudy 15  
## 3 2017-9-2 Rainy 18

colnames(weather)

## [1] "date" "sky" "temp"

weather$sky

## [1] Sunny Cloudy Rainy   
## Levels: Cloudy Rainy Sunny

weather$sky==weather[,2]

## [1] TRUE TRUE TRUE

## Data Structure - 3. data.frame (2/2)

class(weather$date)

## [1] "factor"

weather$date <- as.Date(weather$date)  
weather$sky <- as.character(weather$sky)  
weather

## date sky temp  
## 1 2017-08-31 Sunny 20  
## 2 2017-09-01 Cloudy 15  
## 3 2017-09-02 Rainy 18

sapply(weather, class)

## date sky temp   
## "Date" "character" "numeric"

## Data Structure - 4. list

drwon <- list(numJobs = c(1,3),  
 car = c("Mercedez","Porche","Kia"),  
 hobby = c("golf", "squash"))  
drwon

## $numJobs  
## [1] 1 3  
##   
## $car  
## [1] "Mercedez" "Porche" "Kia"   
##   
## $hobby  
## [1] "golf" "squash"

names(drwon)

## [1] "numJobs" "car" "hobby"

drwon[[1]]

## [1] 1 3

drwon$numJobs \* c(10,5)

## [1] 10 15

## Data Structure - Summary

## Data Type - 4. factor

범주형 자료 (Categorical Data)

Group별 특성을 표현 ex) Learning Spoons: Finance, Programming, Data Science,… ex) GICS 산업 분류, Industry grouping, 산업별 수익률 평균 등 String 변수와의 차이점

String & Factor \ message & classification, category, group\ name, 자체로 의미 & attribute, 다른 group과의 차이로서의 의미 \ 일반적으로 unique & 일반적으로 많은 비슷한 것들이 있음. \

## Data Type - 5. Date

mydates <- as.Date(c("2007-06-22", "2004-02-13"))  
mydates[1] - mydates[2]

## Time difference of 1225 days

today <- Sys.Date( )  
today

## [1] "2018-05-20"

as.numeric(substr(today,1,4)) # year # substr works for date

## [1] 2018

as.numeric(substr(today,6,7)) # month

## [1] 5

as.numeric(substr(today,9,10)) # day

## [1] 20

format(today, format="%B %d %Y")

## [1] "5월 20 2018"

## lubridate package

## Date Type - 일반적인 방법

String, Factor, Date간의 변환

R에서는 data.frame을 생성할 때 문자로된 데이터를 자동으로 **factor**로 처리함.

1. 따라서 stringsAsFactors=FALSE의 option을 data.frame 생성시 추가! (그리고 factor 변수로 원할때에 as.factor()를 사용하여 변환)
2. 날짜인데 String으로 분류된 변수들에 대해서 as.Date()를 이용하여 type을 변환!

## Date Type

* 데이터 타입은 총 몇개인가요? 이것들을 다 외어야 하나요??
* 지금까지 커버된 5개 데이터 타입은 자주 쓰이고 중요함
* 그러나 데이터 타입은 무한대…
* 심지어 새로운 패키지마다 그 패키지의 클래스(class)에 적합한 데이터 타입을 나름대로 정의하기도 함
* 마주칠 때마다 검색하여 *대충* 필요한 만큼 알아보고 필요한 부분만 사용하는 것이 프로그래밍

## 실습 과제

function <- tree(h) {  
 ...  
 ...  
 ...  
 ...  
 ...  
}

execute

tree(4)

## [1] " A "  
## [1] " AAA "  
## [1] " AAAAA "  
## [1] "AAAAAAA"

tree(6)

## [1] " A "  
## [1] " AAA "  
## [1] " AAAAA "  
## [1] " AAAAAAA "  
## [1] " AAAAAAAAA "  
## [1] "AAAAAAAAAAA"