Lab practice2

PR2\_201821479\_황혜린

2018년 9월 13일

# 1.R로 계산하기

## 1.1 기본연산

31 + 3

## [1] 34

15 - 3 + 7

## [1] 19

13 \* 2 - 6 / 2

## [1] 23

13 \* (2 - 6) / 2

## [1] -26

8 %/% 2

## [1] 4

11 %% 3

## [1] 2

n <- 21 %% 4  
print (n)

## [1] 1

## 1.2 수학함수 사용

log(2)

## [1] 0.6931472

log(exp(1))

## [1] 1

sqrt(4)

## [1] 2

4^5

## [1] 1024

4\*\*5

## [1] 1024

round(9.13)

## [1] 9

ceiling(1.41)

## [1] 2

floor(1.95)

## [1] 1

pi

## [1] 3.141593

# 2. 수치 요약하기

## 2.1 벡터 생성 및 출력

* 정수형 값이 저장된 벡터를 생성하기
* 벡터 출력해보기

v1 <- 3  
v2 <- c(4,5)  
v3 <- 3:11  
v4 <- c(v1,v2,v3)  
print(v1)

## [1] 3

print(v2)

## [1] 4 5

print(v3)

## [1] 3 4 5 6 7 8 9 10 11

print(v4)

## [1] 3 4 5 3 4 5 6 7 8 9 10 11

## 2.2 평균구하기

* 평균을 구하는 여러가지 방법

(1+2+3+4+5+6+7+8+9) / 9

## [1] 5

sum(1+2+3+4+5+6+7+8+9) / 9

## [1] 5

v5 <- 1:9  
sum(v5) / length(v5)

## [1] 5

mean(v5)

## [1] 5

## 2.3 함수활용

mean(v5)

## [1] 5

var(v5)

## [1] 7.5

sd(v5)

## [1] 2.738613

median(v5)

## [1] 5

max(v5)

## [1] 9

min(v5)

## [1] 1

v6 <- 1:10  
median(v6)

## [1] 5.5

# 3. 문자값이 저장된 벡터 생성

myEmail <- "yny0506@naver.com"  
birthday <- c("1994 년" , "5월" , "6일")  
birthday2 <- paste("1994 년" , "5월" , "6일")  
birthday3 <- paste0("1994 년" , "5월" , "6일")  
  
print(myEmail)

## [1] "yny0506@naver.com"

print(birthday)

## [1] "1994 년" "5월" "6일"

print(birthday2)

## [1] "1994 년 5월 6일"

print(birthday3)

## [1] "1994 년5월6일"

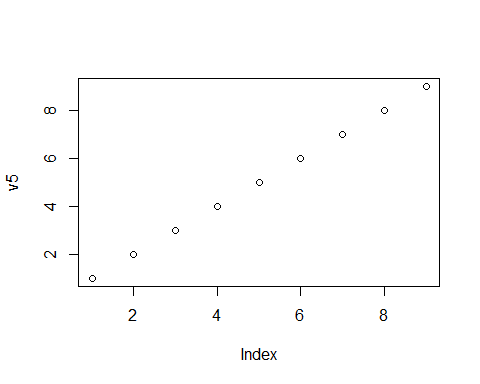
# 4. 기타 R활용법

## 4.1 세미콜론 활용 및 변수명만으로 출력하기

mean(v5) ; sd(v5) ; plot(v5)

## [1] 5

## [1] 2.738613



myEmail ; birthday

## [1] "yny0506@naver.com"

## [1] "1994 년" "5월" "6일"

## 4.2 작업폴더 확인 및 변경

* 변경할 폴더는 사전에 만들어져 있는 폴더여야함
* 본인이 작업할 폴더의 경로를 setwd(“”)의 따옴표 사이에 입력
* 작업할 폴더는 본인이 원하는 경로로 지정해주세요.
* 예)setwd(“c:/data”)

getwd()

## [1] "C:/Users/hyere/Desktop"

setwd("C:/Users/hyere/Desktop/R")  
getwd()

## [1] "C:/Users/hyere/Desktop/R"

# 1. R에서 기초적인 4가지 Data Type

num <- 3 ; class(num)

## [1] "numeric"

numVec <- c(1,2,3) ; class(numVec)

## [1] "numeric"

comp <- 2 + 3i ; class(comp)

## [1] "complex"

compVec <- c(2 + 3i, 4 + 5i, 6 + 7i) ; class(compVec)

## [1] "complex"

char <- "a" ; class(char)

## [1] "character"

charVec <- c("a","b","c") ; class(charVec)

## [1] "character"

logic1 <- TRUE ; class(logic1)

## [1] "logical"

logic2 <- T ; class(logic2)

## [1] "logical"

logic3 <- FALSE ; class(logic3)

## [1] "logical"

logic4 <- F ; class(logic4)

## [1] "logical"

# 2. numeric data와 complex data의 연산

comp + num

## [1] 5+3i

comp - num

## [1] -1+3i

comp \* num

## [1] 6+9i

comp / num

## [1] 0.666667+1i

comp \* 1i

## [1] -3+2i

log(comp)

## [1] 1.282475+0.982794i

sqrt(comp)

## [1] 1.674149+0.895977i

# 3. vector간의 연산

vec1 = c(2,4,1,3,4,5,1,2,3,5)  
vec2 = c(4,5,2,3,8,3,4,1,5,2)  
  
vec1 + vec2

## [1] 6 9 3 6 12 8 5 3 8 7

vec1 - vec2

## [1] -2 -1 -1 0 -4 2 -3 1 -2 3

vec1 \* vec2

## [1] 8 20 2 9 32 15 4 2 15 10

vec1 / vec2

## [1] 0.500000 0.800000 0.500000 1.000000 0.500000 1.666667 0.250000  
## [8] 2.000000 0.600000 2.500000

vec1 > vec2

## [1] FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE

vec1 >= vec2

## [1] FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE

12 + vec1

## [1] 14 16 13 15 16 17 13 14 15 17

12 / vec1

## [1] 6.0 3.0 12.0 4.0 3.0 2.4 12.0 6.0 4.0 2.4

# 4. character data

char1 <- "a" ; class(char1)

## [1] "character"

char2 <- "character" ; class(char2)

## [1] "character"

char3 <- "year: 2018" ; class(char3)

## [1] "character"

char4 <- "3" ; class(char4)

## [1] "character"

# 5. logical data

logic1 <- 4>5 ; logic1 ; class(logic1)

## [1] FALSE

## [1] "logical"

logic2 <- 7>2 ; logic2 ; class(logic2)

## [1] TRUE

## [1] "logical"

logic3 <- 1:9 ; logic3 ; class(logic3)

## [1] 1 2 3 4 5 6 7 8 9

## [1] "integer"

logic4 <- c(T,F,F,T,F,T,F,T,T,F) ; logic4 ; class(logic4)

## [1] TRUE FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE FALSE

## [1] "logical"

T + T

## [1] 2

TRUE \* FALSE

## [1] 0

sum(T, T, F, T, F)

## [1] 3

# 6. Special Value

-Inf \* 3

## [1] -Inf

# 7. 수열 생성하기

1:9

## [1] 1 2 3 4 5 6 7 8 9

seq(from=1, to=9, by=1)

## [1] 1 2 3 4 5 6 7 8 9

seq(from=1, to=9, by=3)

## [1] 1 4 7

seq(1, 9, 3)

## [1] 1 4 7

seq(1,9,length.out=3)

## [1] 1 5 9

# 8. 원소들을 반복하여 사용하기

rep(c(1,2,3), each=4)

## [1] 1 1 1 1 2 2 2 2 3 3 3 3

rep(c(1,2,3), time=4)

## [1] 1 2 3 1 2 3 1 2 3 1 2 3

rep(c(1,2,3), each=4, time=4)

## [1] 1 1 1 1 2 2 2 2 3 3 3 3 1 1 1 1 2 2 2 2 3 3 3 3 1 1 1 1 2 2 2 2 3 3 3  
## [36] 3 1 1 1 1 2 2 2 2 3 3 3 3

# 9. 벡터에 저장된 값 추출 및 수정

a = 1:9 ; a

## [1] 1 2 3 4 5 6 7 8 9

a[1]

## [1] 1

a[1:4]

## [1] 1 2 3 4

a[c(1,2,5)]

## [1] 1 2 5

a[c(-2:-4)]

## [1] 1 5 6 7 8 9

a[a > mean(a)]

## [1] 6 7 8 9

a[a == mean(a)]

## [1] 5

a[a < mean(a)] = 1 ; a

## [1] 1 1 1 1 5 6 7 8 9

append(a,10) ; a

## [1] 1 1 1 1 5 6 7 8 9 10

## [1] 1 1 1 1 5 6 7 8 9

append(a,10,2) ; a

## [1] 1 1 10 1 1 5 6 7 8 9

## [1] 1 1 1 1 5 6 7 8 9

sort(a,decreasing=T)

## [1] 9 8 7 6 5 1 1 1 1

sort(a,decreasing=F)

## [1] 1 1 1 1 5 6 7 8 9

order(a,decreasing = T)

## [1] 9 8 7 6 5 1 2 3 4

order(a,decreasing = F)

## [1] 1 2 3 4 5 6 7 8 9

# 10. 통계함수

a

## [1] 1 1 1 1 5 6 7 8 9

mean(a)

## [1] 4.333333

var(a)

## [1] 11.25

sd(a)

## [1] 3.354102

summary(a)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 5.000 4.333 7.000 9.000

# 9. 기타 벡터 다루기

object.size(a)

## 176 bytes

length(a)

## [1] 9

nchar("alphago")

## [1] 7

length("alphago")

## [1] 1

letters[1:5]

## [1] "a" "b" "c" "d" "e"

names(a) = c("c1", "c2", "c3", "c4", "c5", "c6") ; a

## c1 c2 c3 c4 c5 c6 <NA> <NA> <NA>   
## 1 1 1 1 5 6 7 8 9

# PR2 연습문제

## 문제 1.R기본연산

round(log(3) + sqrt(9))

## [1] 4

round(log(3)) + sqrt(9)

## [1] 4

x = 20:29  
mean(x)

## [1] 24.5

sd(x)

## [1] 3.02765

median(x)

## [1] 24.5

setwd("C:/Users/hyere/Desktop/R\_programming") #작업 폴더 변경  
getwd()

## [1] "C:/Users/hyere/Desktop/R\_programming"

## 문제 4.Vector

vec1 <- seq(0, 100, 2)  
  
length(vec1) #벡터에 있는 data개수

## [1] 51

append(vec1, 10000, 26) #현재 vec1에서 vec1의 평균값은 vec1에 할당한 data개수의 가운데 값에 존재합니다. 따라서 앞에서 구한 51의 가운데 값인 26을 append의 마지막 값에 삽입했습니다.

## [1] 0 2 4 6 8 10 12 14 16 18 20  
## [12] 22 24 26 28 30 32 34 36 38 40 42  
## [23] 44 46 48 50 10000 52 54 56 58 60 62  
## [34] 64 66 68 70 72 74 76 78 80 82 84  
## [45] 86 88 90 92 94 96 98 100

vec1[vec1 > mean(vec1)]

## [1] 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84  
## [18] 86 88 90 92 94 96 98 100

num1 <- sort(vec1,decreasing=T) #num1에 vec1의 내림차순 입력  
print(num1) #프린트 num1

## [1] 100 98 96 94 92 90 88 86 84 82 80 78 76 74 72 70 68  
## [18] 66 64 62 60 58 56 54 52 50 48 46 44 42 40 38 36 34  
## [35] 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2 0

day <- 1:7 #day에 요일벡터 숫자로 할당  
names(day) = c("평일", "평일", "평일", "평일", "평일", "주말", "주말") ; day #각 요일벡터에 평일, 주말 이름을 붙임

## 평일 평일 평일 평일 평일 주말 주말   
## 1 2 3 4 5 6 7

rep(day, time = 4) #그것을 1\*4주분량으로 반복

## 평일 평일 평일 평일 평일 주말 주말 평일 평일 평일 평일 평일 주말 주말 평일   
## 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1   
## 평일 평일 평일 평일 주말 주말 평일 평일 평일 평일 평일 주말 주말   
## 2 3 4 5 6 7 1 2 3 4 5 6 7

## 도전문제1.

prices = c(11905000, 11973000, 12190000, 12700000, 12303000, 12604000)  
prices\_today = c(11973000, 12190000, 12700000, 12303000, 12604000)  
prices\_yesterday = c(11905000, 11973000, 12190000, 12700000, 12303000)  
  
x = prices\_today / prices\_yesterday #{(금일의 종가-전일의 종가)/전일의 종가} = {(금일의 종가/전일의 종가)-1} 입니다. 여기서 x= 금일의 종가/전일의 종가 입니다  
y = (x - 1) \* 100 #수익률   
names(y) = c("2018-03-02" ,"2018-03-03", "2018-03-04", "2018-03-05", "2018-03-06") ; y #수익률에 날짜 할당

## 2018-03-02 2018-03-03 2018-03-04 2018-03-05 2018-03-06   
## 0.5711886 1.8124113 4.1837572 -3.1259843 2.4465578

returns <- y #구한 값, 즉 수익률y를 returns란 변수에 저장

## 도전문제2.

mean(returns)

## [1] 1.177586

var(returns)

## [1] 7.484698

returns[returns < mean(returns)]

## 2018-03-02 2018-03-05   
## 0.5711886 -3.1259843

z<- sort(returns,decreasing=F) #수익률을 오름차순으로 정리하고 z에 저장해 둡니다.  
min <- z[1] #최소수익률=오름차순의 첫번째data  
print(min)

## 2018-03-05   
## -3.125984

max <- z[5] #최대수익률=오름차순의 마지막data  
print(max)

## 2018-03-04   
## 4.183757