

IntroR.Note

Sung Won Kang

2016년 10월 29일

1주차: R 기초

scalar 와 벡터

1. 설치가 잘 되었을까?

```
#summary(cars)
help(apply)
?apply
?help.search
example(apply)
```

2. 메모리 점검

- 우리는 어디에 있을까?

```
getwd()
```

- 지금 메모리에 떠 있는 것은?

```
objects()
ls()
```

- 메모리를 비우려면 ? rm

```
list.ls=ls()
rm(list=list.ls[1])
ls()
rm(list=ls())
ls()
```

- Tip: R은 대소문자를 구분

```
#HELP(apply)
```

3. Scalar

i. 만들기

```
#Four types. Numeric, character, logic, factor
#숫자(numeric)
x1=10
x2=2.3
x3=3
#문자(string)
S1="Hello world"
S2="My name is ... "
#논리연산자(logic)
L1=TRUE
L2=FALSE
```

ii. 연산

```
x1+x2
x1-x2
x1/x2
x1*x2
x1^x2
x1/%x3 #몫
x1 %% x3 #나머지
x1/%x2 #몫
x1 %% x2 #나머지

nchar(S1)# 문자 수
paste(S1,S2) # 붙이기

L1|L2 # 둘 중 하나는 맞아요
L1 & L2 # 둘 다 맞아요
! L1 # L1은 아니에요
! L2
```

4. 벡터: 간단한 조작

i. 벡터 만들기 1: 숫자 집어넣기

```
x=c(10.4, 5.6, 3.1, 6.4, 21.7)
x
xn=(1:10)
xn
xr=rep(c(0,2),5)
xr
```

ii. 벡터와 숫자(Scalar) 연산: 쉬운것부터

```
x+2
x-2
x^3
1/x
```

iii. 벡터 늘리기

```

y=c(x,0)
y
y=c(x,0,y)
y
y3=rep(x,10)
y3

```

iv. 벡터 연산: 벡터와 벡터

a. 서로 다른 길이의 벡터를 더하면? 길이가 늘어남(recycle) *칠판에 써서..

```

v1=2*x+4
v1

v2=2*x+y+1
v2

v3=v1+v2
v3

length(x)
length(y)
length(v1)
length(v2)
length(v3)

```

b. 벡터 연산 2.

```

x
v1
x+v1
x-v1
x*v1
x/v1
x^v1

```

길이가 다르면?

```

x/y
x^y

```

c. 벡터연산 3. 함수

```

#로그
log(x)
#초월함수
exp(x)
#삼각함수
sin(x)
cos(x)
tan(x)
#지수함수
sqrt(x)
#길이
length(x)
#합, 곱
sum(x)
prod(x)
#표본통계치
max(x)
min(x)
range(x)
mean(x)
var(x)
sd(x)
median(x)
#정렬
sort(x)
sort(x,decreasing=T)

```

만약 두 개를 넣으면?

```

## 뱌어내거나
#초월함수
#exp(x,v1)
#길이
#length(x,v1)
#정렬
#sort(x,v1)
# 평균
#mean(x,v1)

## 넁넁거리거나
#합, 곱
sum(x,v1)
#표본통계치
max(x,v1)
min(x,v1)
range(x,v1)
var(x,v1)

## 첫번째 것만 쓰거나
sd(x,v1)
median(x,v1)

```

v. 벡터 만들기 2: 수열,논리,문자

w. 수열

```

xn=(1:10)
xn
## 등차수열
# 항의 수를 모르는 경우
xseq=seq(1,9,2)
xseq
xseq.1=seq(from=1,to=9,by=2)
# 마지막 값을 모르는 경우
xseq.2=seq(from=1,length=5,by=2)
xseq.2
# 등차를 모르는 경우
xseq.3=seq(from=1,to=9,length=5)
xseq.3

## 등비수열?
## 초항 2. 등비 2. 항수 10 등비수열
xseqp=2^xn
xseqp
## 초항 100. 등비 2. 항수 10 등비수열
xseqp2=100*(2^(0:9))
xseqp2
## 초항 100, 등비 1/2. 항수 10 등비수열
xseqp3=100*((1/2)^(0:9))
xseqp3

```

b. 논리 : 일치. 불일치.대소관계. and/or

```

xeq=(x == 4)
xeq

xneq=(x != 4)
xneq

xge=(x >= 4)
xge

xgt=(x > 4)
xgt

xge=(x >= 4)
xge

xle=(x <= 4)
xle

xlt=(x < 4)
xlt

xand=(xgt & xge)
xand
xge

xor=(xgt | xlt)
xor
xneq

```

c. 문자 그리고 paste

```
xs=c("1","2","3","4")
#xsa=c(A,B,C,D)
#xsa
xsa=c("A","B","C","D")
xsa
```

- 문자열을 붙일때

```
paste(xsa, collapse="")
paste(xsa, collapse=" ")
paste(xsa, collapse="-")
```

- 문자열에 1렬번호를 매길때

```
paste(xsa,(1:5), sep="")
paste(xsa,(1:5), sep=" ")
paste("X",(1:5), sep="-")
```

vi. 번호매개기(index), 고르기

a. 비어있음(missing value: NA), 숫자가 아님(NaN)

```
z=c((-10:10),NA)
z
is.na(z)
z1=0/0
z1
is.na(z1)
is.nan(z)
is.nan(z1)
```

b. 벡터 중 일부만 고르기

- 조건문으로 고르기

```
#NA 가 아닌것
z[!is.na(z)]
#0 보다 큰 것
z[z>0]
#0 보다 크고 NA가 아닌 것
z[z>0&!is.na(z)]
z+1
(z+1)[z>0&!is.na(z)]
```

- 숫자로 지정

```
# 홀수항, 짝수항
z1=length(z)
z[seq(1,z1,2)]
z[seq(0,z1,2)]
# 내맘대로
z[c(1,11,4)]
# 내 맘대로 빼고
z[-c(1,4,14)]
```

- 문자로 지정

```
fruit=c(5,1,10,10,20,20)
names(fruit)=c("OR","BA","AP","AP","PE","PE")
fruit[c("OR","BA")]
fruit[c(1,2)]
```

- 골라서 바꾸기

```
#NA는 0으로
z[is.na(z)]=0
z
#0 보다 작으면 100으로
z[z<0]=100
z
```

vii. type 바꾸기. 범주 변수

a. type 바꾸기

```
str(fruit)
#문자열로
fruit.s=as.character(fruit)
#sum(fruit.s)
str(fruit.s)
nchar(fruit.s)
paste(fruit.s,collapse="-")
#다시숫자로
fruit.n=as.numeric(fruit.s)
str(fruit.n)
sum(fruit.n)
nchar(fruit.s)
paste(fruit.n,collapse="-")
```

- 범주 변수

```

state = c("tas", "sa", "qld", "nsw", "nsw", "nt", "wa", "wa",
"qld", "vic", "nsw", "vic", "qld", "qld", "sa", "tas",
"sa", "nt", "wa", "vic", "qld", "nsw", "nsw", "wa",
"sa", "act", "nsw", "vic", "vic", "act")

incomes = c(60, 49, 40, 61, 64, 60, 59, 54, 62, 69, 70, 42, 56,
61, 61, 61, 58, 51, 48, 65, 49, 49, 41, 48, 52, 46,
59, 46, 58, 43)

statef=factor(state)
levels(statef)

# 조건부 표본통계량
incmeans=tapply(incomes,statef,mean)
incster=tapply(incomes,statef,sd)
incfreq=tapply(incomes,statef,length)
table(statef)

incomef=factor(cut(incomes, 4))
table(statef, incomef)

incomeh=factor(cut(incomes,2))
table(statef, incomef, incomeh)

```

- 범주 변수의 type 변환

```

as.character(statef)
as.numeric(statef)

fruit.f=as.factor(fruit)
as.numeric(fruit.f)

```

Array, matrix (multi-dimension)

1. Array

```

z=(1:24)
z
str(z)

dim(z)=c(3,4,2)
z
str(z)
z[1:4]
z[1,,]
z[,1,]
z[, ,1]

```

2.matrix= Array with two subscripts


```

dim(z)=c(8,3)
z
str(z)
z[1:4]
z[1,]
z[,1]

A=array(incomes[1:24],c(8,3))

## 행렬계산
A+z
A-z
A*z

t(A)%*%z

C=(1:3)

A%*%C

#inner product
t(C)%*%C
#outer product
C%O%C
# 대각행렬
DC=diag(C)
#역행렬
AA=t(A)%*%A
solve(AA)
AA%*%solve(AA)
# 선형연립방정식
y=AA%*%C
solve(AA,y)

#identity matrix
ID=diag(3)
AA%*%ID-AA

#zero matrix
nr=3
nc=3
mzero=rep(0,nr*nc)
dim(mzero)=c(nr,nc)

str(z)
str(A)
str(AA)
str(DC)
str(ID)
str(mzero)

```

3. matrix from vector,vector from matrix

```
X1=c(1,2,3)
X2=c(4,100,20)
XX=cbind(1,X1,X2)
str(XX)

as.vector(XX)
c(XX)
```

List and dataframe

1. list : 아무거나 다 넣는 공간

```
Lst=list(name="Fred",wife="Mary", no.children=3,child.ages=c(4,7,9))
length(Lst)
names(Lst)
str(Lst)
Lst[[1]]
Lst[1]
str(Lst[[1]])
str(Lst[1])
sum(Lst[[4]])
#sum(Lst[4])
Lst2=list(name="강성원",wife="?", no.children=2,child.ages=c(2,9))
LstLong=c(Lst,Lst2)
str(LstLong)
```

2. data frame : NxK 형 저장공간

```
DA=data.frame(AA)
Stateincome=data.frame(state,incomes)

str(Stateincome)
summary(Stateincome)
attach(DA)
detach()
```

3. data frame과 vector

```
X1=(1:10)
X2=(101:110)
X3=(1001:1010)

D=data.frame(X1,X2,X3)
DS_C=apply(D,2,sum)
print("D/DS_C=")
print(D/DS_C)
# 합쳐서 1이 안되네?

print("t(t(D)/DS_C=")
print(t(t(D)/DS_C))
# 이제 합쳐서 1 되네?

a=c(10,100,1000)
print("a=")
print(a)
print("D/a=")
print(D/a)

print("t(t(D)/a)")
print(t(t(D)/a))
divby.a=data.frame(t(t(D)/a))
print("divided by a_columnwise")
print(divby.a)

DS_R=apply(D,1,sum)
Share_row=D/DS_R
print("divison by row sum")
print(Share_row)
```

4. data frame에 쓰는 함수들

```
#표본통계치
colSums(D)
rowSums(D)
colMeans(D)
rowMeans(D)
dim(D)

#다른 표본통계치는? apply
makez=function(x) (x-mean(x))/sd(x)
cv=function(x) sd(x)/mean(x)
apply(D,1,FUN=makez)
apply(D,2,FUN=makez)

#조건부 표본통계치 aggregate
str(iris)
aggregate(data.frame(iris[,1:4]),by=list(iris$Species),FUN=mean)
aggregate(cbind(Sepal.Width,Sepal.Length)~Species,data=iris,FUN=mean )
aggregate(~Species,data=iris,FUN=mean )
aggregate(~Species,data=iris,FUN=cv)

aggregate(~state,data=Stateincome,FUN=mean)

# 줄세우기(order)
Stateincome[order(Stateincome[,1],Stateincome[,2]),]
Stateincome[order(Stateincome[,1],Stateincome[,2],decreasing=T),]
```