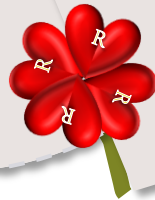


7. 함수(function)

프로그램의 간결성을 위해 함수를 이용합니다.
내장함수 종류와 사용자정의함수 작성방법을
학습합니다.



함수(function)?

음식과 운동량 프로그램



나 달리기 -30분 했어.
칼로리좀 계산해줘



멍멍짖기 -30분 칼로리좀
계산해줘



i'm calorie calculator!!
칼로리 계산부분은 나를
호출하세요^^

•함수(function) - 특정한 작업을 독립적으로 수행하는 프로그램 코드의 집합체

I.R 에서 많이 사용하는 기본함수

함수	기능	함수	기능
mean()	평균값을 계산	min(),max()	최소값,최대값을 계산
sum()	합계를 계산	median()	중위수 계산
sd()	표준편차를 계산	cor()	상관계수 계산
var()	분산 계산	range()	범위를 계산
sort()	정렬작업 수행	rev()	요소의 나열순서를 반대로 표시



```
> a<-c(33,12,55,66,77,16)
> rev(a)
[1] 16 77 66 55 12 33
> mean(a)
[1] 43.16667
> sort(a)
[1] 12 16 33 55 66 77
> range(a)
[1] 12 77
> median(a)
[1] 44
> sd(a)
[1] 26.90291
> var(a)
[1] 723.7667
> sum(a)
[1] 259
```

□ apply()함수

- 벡터, 행렬, 데이터프레임 등에서 행/열 단위의 필요한 계산작업을 쉽게 하는데 유용한 함수
- 형식 apply(대상객체, 행 또는 열, 적용할함수)



```
> height<-c(177,180,167,165,170,NA)
> weight<-c(NA,77.3,80,60,NA,64)
> df<-data.frame(신장=height, 체중=weight)
> names(df)<-c("신장","체중")
> df
  신장 체중
1  177   NA
2  180 77.3
3  167 80.0
4  165 60.0
5  170   NA
6   NA 64.0
```

위와 같은 데이터프레임에서 결측값(NA) 제외한 컬럼별(신장,체중) 평균을 계산하는 작업

```
> apply(df,2,mean,na.rm=TRUE)
  신장   체중 
171.800  70.325 
> |
```

```
> lapply(df,max,na.rm=TRUE)
$신장
[1] 180

$체중
[1] 80

> sapply(df,max,na.rm=TRUE)
신장 체중 
180  80 
> |
```

apply()에서 더 발전한 함수들을 살펴보죠.
 lapply() - 계산결과를 list형태로 변환하여 출력
 sapply() - 계산결과를 벡터형태로 변환하여 출력
 tapply() - 계산결과를 특정부분으로 그룹화하여 출력

```
> dflist<-read.table("d:\\r1\\passlist.txt")
```

```
> dflist
```

	V1	V2	V3
1	나이	성별	평점
2	33	남	4.3
3	28	남	4.2
4	35	여	4.1
5	29	여	3.7
6	36	남	4.5
7	32	여	4.4
8	30	여	3.8

tapply() 이용하여 성별 평점을 계산

```
> tapply(dflist$평점,dflist$성별,mean)
```

남	여
4.181429	4.163846

tapply(벡터,인덱스,사용할함수)

2. 사용자 정의함수(function)

- 프로그램에 반복적으로 사용되는 특별한 계산식을 직접 작성
- 함수 생성형식

함수이름 <- function(인자) {함수의 몸체 ..return(반환값)}

인자 : 함수가 사용할 인수들(argument)-없을 수도 있어요.
함수의 몸체 : 함수가 호출되면 실행하게 될 프로그램 본문
함수이름 : 정의된 함수를 할당하여 생성되는 객체이름.

□ 인자가 없는 가장 기본적인 함수 예제

```
> guideinfo<-function(){  
+ writeLines("분석을 위한 안정된 프로그램 R")  
+ }  
> guideinfo() 함수호출  
분석을 위한 안정된 프로그램 R
```

함수몸체에서는 “분석을 위한 안정된 프로그램 R”을 출력하게끔 설계됨

□ 인자가 있는 함수 생성

```
> guideinfo<-function(msg) {
+ writeLines(msg)
+ }
> guideinfo("분석작업 오케이")
분석작업 오케이
> guideinfo("내 마음대로!!")
내 마음대로!!
```

msg ← 인자, 여기에 값을 전달해주면
writeLines 함수 실행하여 결과를 보여주는거죠^^

□ 사용자정의함수 실습작업

```
> basicst<-function(x) {
+ amin<-min(x)
+ amax<-max(x)
+ amean<-mean(x)
+ astd<-sd(x)
+ avar<-var(x)
+ totinfo<-list(최소값=amin, 최대값=amax, 평균=amean, 표준편차=astd, 분산=avar)
+ return(totinfo)
+ }
> a<-c(68,75,80,85,92,77,83,85,95)
> basicst(a)
$최소값
[1] 68

$최대값
[1] 95

$평균
[1] 82.22222

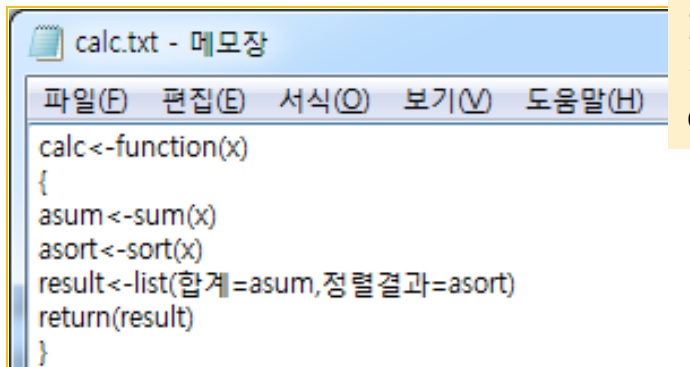
$표준편차
[1] 8.378212

$분산
[1] 70.19444
```

basicst함수소개

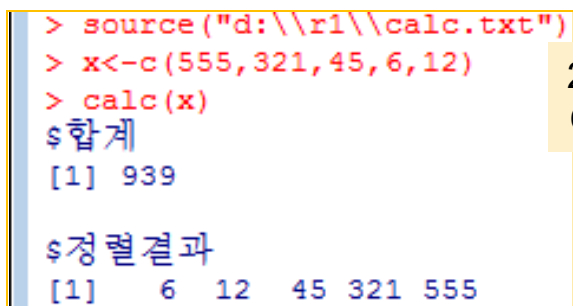
basicst함수에 인자만 입력하면
값 집단의 최소, 최대, 평균값은 물론이고,
분산, 표준편차까지 모두 계산하는 함수 입니다.

□ 외부파일에 정의한 함수 사용하기



```
calc.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
calc<-function(x)
{
  asum<-sum(x)
  asort<-sort(x)
  result<-list(합계=asum,정렬결과=asort)
  return(result)
}
```

1. 메모장에서 그림처럼 값의 합계와 정렬을 실행하는 calc 함수를 생성하고 저장합니다.
calc.txt ← 여러분의 작업디렉토리에



```
> source("d:\\r1\\calc.txt")
> x<-c(555,321,45,6,12)
> calc(x)
$합계
[1] 939

$정렬결과
[1] 6 12 45 321 555
```

2. source함수를 이용하여 외부에서 정의한 함수파일을 가져옵니다.
이 작업이 이뤄지면 여러분은 calc함수를 사용할 수 있게 됩니다.