제 9장 함수

1. 사용자 정의함수: 사용자가 만든 함수

```
함수이름 <- function(인수1, 인자2, 인자3,…){  # 인자는 없어도 됨.
                                계산과정
                                 return(결과) # 결과는 1개만
            }
# 인자, return(결과)는 없어도 됨.
```

eg1) 함수를 실행하면 "Hello !" 가 프린트되는 이름이 'greet' 라는 함수 만들기

```
greet<-function(){
      print("Hello !")
      }
greet()
```

eg2) 함수에 x를 입력하여 실행하면 'x가 7의 배수인지 아닌지'의 결과가 구해지는 이름이 'seven' 이라는 함수 만들기

```
seven<-function(x){
   if(x%%7==0){cat(x,"=7의 배수")
   } else {cat(x,"=7의 배수아님") }
}
seven(114)
```

eg3) 함수에 x를 입력하여 실행하면 'x<0 이면 0, x≥0이면 x'의 결과가 구해지는 이름이 'relu' 라는 함수 만들기

```
relu<-function(x){
    if (x<0) {return(0)
    } else {return(x)}
}
relu(-2.1); relu(1.5)
```

eg4) 함수에 x1,x2을 입력하여 실행하면 '100*(x1+x2)'의 결과가 구해지는 이름이 'hundr' 이라는 함수 만들기

```
hundr<-function(x1,x2){
    x=100*(x1+x2)
    return(x)    # = return(100*(x1+x2)
}
hundr(1, 3)
y=hundr(2.1, 2.4) # hundr(2.1, 2.4) 결과를 y에 저장
```

eg5) 숫자 x를 입력하여 실행하면. x-7, x+7이 구해지는 이름이 'seven2' 이라는 함수 만들기.

```
seven2<-function(x){
  y=numeric(2)
  y[1]=x-7
  y[2]=x+7
return(y)
}

z=seven2(8)
z
z[1]
z[2]
```

2. 내장함수

R에 설치되어 있는 함수

```
log(2); log10(12)
# 표본추출 함수
sample(x, n, replace=T, prob=p)
# x:모집단, n:표본크기,  replace=T: 복원추출(없으면 비복원),
# prob=p: x의 각원소가 표본에 뽑힐 확률(없으면 모두 동일)
s1=sample(1:10, 3)
s1
mean(s1); sd(s1)
```

eg6) 함수형 수치형벡터 x를 입력하여 실행하면 평균과 분산이 구해지는 이름이 'summr' 이라는 함수 만들기

```
 summr<-function(x){
     y=numeric(2) # 크기가 2인 벡터 만들기.
     y[1]=mean(x); y[2]=var(x)
     return(y)
 }
```

제 10장 통계

1. 확률분포와 통계량
2. 정수 모의실험
4. 통계분석

1. 확률분포와 통계량

(1) 도수분포표

```
# 범주형자료: 10명의 선호도 결과 : (저,중,상)=(L, M, H)
x<-c('L' ,'H', 'M', 'M', 'L', 'H', 'H', 'M', 'H', 'M')
table(x) # 도수
prop.table(table(x)) # 상대도수
# 수치형자료:
```

(2) 랜덤표본(random sample)

$X \sim N(\mu,\sigma^2)$: 확률변수 X는 평균이 μ이고 표준편차가 σ>0 (분산이 σ^2)인 정규분포를 따른다.

확률변수 X의 n개의 실제값을 생성한 것 = $X \sim N(\mu,\sigma^2)$를 따르는 n개의 난수

```
x<-rnorm(1000, 0,1.5) # N(0,1.5^2)를 따르는 난수 1000개 생성
mean(x); sd(x)
# 생성된 난수가 정규분포에서 생성된 것인지 확인, 직선모양~ 정규분포에서 생성
qqnorm(x)
```

2. 정수 모의실험

(3) 정수추출 모의실험

1~100사이 정수에서 40개를 비복원 추출

```
x<-sample(1:100, 40) # replace=T ~ 복원 추출

# 추출된 수 중 50이하인 수의 개수, 합&평균
x[x<=50]
length(x[x<=50])
sum(x[x<=50]); mean(x[x<=50])
```

4. 통계분석

(1) 기초통계

```
# summary(): 자료의 요약함수, 수치형: 최소값, 제1사분위수, 중앙값, 제3사분위수, 최대값
```

```
xN<-sample(1:100, 40)
xC<-c('L' ,'H', 'M', 'M', 'L', 'H', 'H', 'M', 'H', 'M')
summary(xN)
table(xC)
```

(4) 단순선형회귀분석

주어진 자료(x,y)의 관계가 $y=\beta_0+\beta_1 x+e, e\sim N(0,\sigma^2)$(단순선형회귀모형)이라고 가정해서 ($\beta_1$: x가 1증가시 y의 증가분) (β_0,β_1)의 추정값($\hat{\beta}_0,\hat{\beta}_1$) 및 추정회귀직선($\hat{y}=\hat{\beta}_0+\hat{\beta}_1 x$)을 주어진 자료(x,y)를 이용해서 구함.

```
x<-1:10
y<-1+1.5*x+rnorm(10,0,1.5)
L=lm(y~x) # 선형회귀분석 함수
L
yh=L$fitted # 추정회귀직선의 ŷ값 = 추정회귀식으로 구한 y의 추정값
plot(x,y, pch='*')
lines(x,yh)
# x=2.5, 4.7 인 경우 y의 예측값
yh=predict(L,newdata=data.frame(x=c(2.5, 4.7)))
```

제 11장 Apply 함수군

주어진 함수 연산을 특정 단위로 쉽게 할 수 있도록 지원하는 함수군

1. apply() 함수

행렬에서 행(Row) 또는 열(Column) 단위의 연산을 쉽게 할 수 있도록 지원하는 함수 - 1이 행 단위 연산이고 2가 열 단위 연산. sum, min, max, prod(곱),mean,...

```
a=matrix(1:12, ncol=4)
a
# 행단위(행별)  합 계산
for(i in 1:3){
print(sum(a[i,])) # 번째 행의 합
}
apply(a,1,sum)
apply(a,2,sum)
apply(a,1,min); apply(a,1,max); apply(a,1,prod); apply(a,1,mean); apply(a,1,sd)
apply(a,1,quantile)
# 결측치(NA) 처리 ~ 빼고 계산
b=a; b[2,3]=NA; b
apply(b,1,sum)
# 결측치(NA) 빼고 계산
apply(b,1,sum, na.rm=T)
```

2. lapply() 함수
리스트에 적용되는 apply() 함수.

```
L=list(x=1:10, y=c(T,F,T,T,F), z=c(3,5,7,11,13,17))
lapply(L, mean)
```

3. sapply() 함수
리스트에 적용되지만 결과가 리스트 형태가 아닌 벡터나 행렬처럼 단순화된 형태로 출력

```
sapply(L, mean)
```

3. vapply() 함수
sapply() 함수처럼 리스트 데이터 객체에 함수를 적용시켜서 함수의 연산 결과를 반환
FUN.VALUE라는 인자를 사용하여 출력되는 결과의 양식을 지정할 수 있다

```
vapply(L, mean,is.logical,FUN.VALUE=0 ) # L의 구성요소가 논리형인가?
vapply(L, sd, FUN.VALUE=double(1)) # L의 구성요소의 평균(T=1, F=0)
```

5. tapply() 함수
범주별(factor)로 함수 적용

```
x=1:12
f=c(rep('A',4), rep('B',4), rep('C',4))
#'A'에 속하는 x의 원소=1,2,3,4,
x
f
tapply(x,f,sum)
```

6. mapply() 함수
여러 개의 인자를 받아 처리하는 함수가 있고 함수에 넘겨줄 인자들이 데이터로 저장되어 있을 때, 데이터에 저장된 값들을 인자로 하여 함수를 호출

```
rnorm(n = 1, mean = 0, sd = 1) # mean=1, sd=1인 정규분포 난수 1개
rnorm(n = 2, mean = 10, sd = 1)
rnorm(n = 3, mean = 100, sd = 1)
mapply(rnorm,  c(1, 2, 3), c(0, 10, 100), c(1, 1, 1))
```

리스트의 열 단위로 불러 함수적용

```
L2=list(x=101:105, y=11:15, z=1:5)
mapply(sum, L2$x, L2$z)
mapply(rep, L2$x, L2$z)
```