

데이터과학을 위한 **R**프로그래밍

2주차. 벡터, 행렬의 연산 및 함수



이혜선 교수

포항공과대학교 산업경영공학과



목차

2주차. 벡터, 행렬의 연산 및 함수

1차시

벡터 및 행렬 생성

2차시

벡터와 행렬의 연산

3차시

간단한 함수 생성 및 루프



2주차

3차시

간단한 함수 생성 및 루프

● 간단한 함수 생성

☑ 함수 생성 (lec2_3.r)

1

square 라는 함수의 생성

프로그램 편집 창

```
# lec2_3.r : loop, for, while
# create a simple function
# square function

square<-function(x){
  return(x*x)
}
square(9)
square(1:3)
```

컨솔창

```
> square<-function(x){
+   return(x*x)
+ }
> square(9)
[1] 81
> square(1:3)
[1] 1 4 9
```

square라는 함수생성됨

square(9)=81

square(1:3)=(1, 4, 9)

● 간단한 함수 생성

✓ 함수 생성

2

dif 라는 함수의 생성

프로그램 편집 창

```
dif<-function(x,y){  
  return(x-y)  
}  
dif(20,10)
```

컨솔창

```
> dif<-function(x,y){  
+   return(x-y)  
+ }  
> dif(20,10)  
[1] 10
```

dif 라는 함수 생성됨

dif(20,10)=10

3

rootdif 라는 함수의 생성

```
rootdif<-function(x,y){  
  return(sqrt(x-y))  
}  
rootdif(20,10)
```

```
> rootdif<-function(x,y){  
+   return(sqrt(x-y))  
+ }  
> rootdif(20,10)  
[1] 3.162278
```

rootdif 라는 함수 생성됨

rootdif(20,10)=3.162278

● 간단한 함수 생성

✓ 기존 함수 : round ← 소수점 자리를 조정

프로그램 편집 창

```
# round off the decimal point  
round(5.14846)  
round(5.14846, 2)
```

```
# to see the function "round"  
round
```

```
round(rootdif(20,10))  
round(rootdif(20,10),2)
```

컨솔창

```
> round(5.14846)  
[1] 5  
> round(5.14846, 2)  
[1] 5.15
```

```
> round  
function (x, digits = 0) .Primitive("round")
```

rootdif(20,10)=3.162278

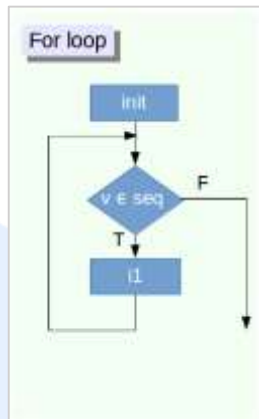
```
> round(rootdif(20,10))  
[1] 3  
> round(rootdif(20,10),2)  
[1] 3.16
```

소수점 둘째 자리로 선택

루프문 (for)

✓ For를 사용한 루프 (예제 1)

➤ 1부터 10까지의 수 중 3으로 나눈 나머지가 1이 아닌 수 출력



for (1에서 10까지)
조건 if (3으로 나누었을 때 나머지가 1)
next (다음숫자로 루프를 돌림)

```
# for 1 to 10
# if remainder=1 when deviding by 3
# then go to next number

for(i in 1:10){
    if(i%%3 == 1){
        next()
    }
    print(i)
}
```

컨솔창

```
> for(i in 1:10){
+   if(i%%3 == 1){
+     next()
+   }
+   print(i)
+ }
[1] 2
[1] 3
[1] 5
[1] 6
[1] 8
[1] 9
```

✓ For를 사용한 루프 (예제 1)

✓ 3으로 나눴을 때의 나머지가 1인 경우 next() → 현재 숫자에서 다음 숫자로 넘어감

✓ 넘어가지 않은 경우 print(i)를 통해 결과를 반환

루프문 (for)

✓ For를 사용한 루프 (예제 2)

➤ 1 부터 10까지 더해가며 합이 20보다 작을 때까지 출력

i=1부터 10까지 1,2,3..으로 더해가며 프린트하는데
sum>20보다 크면 수행(print(sum))를 멈춤

```
# for loop example2
# stop loop after sum>20
sum = 0
for (i in 1:10) {
  sum <- sum + i
  if (sum>20){
    # stop loop after sum>20
    break
  }
  print(sum)
}
```

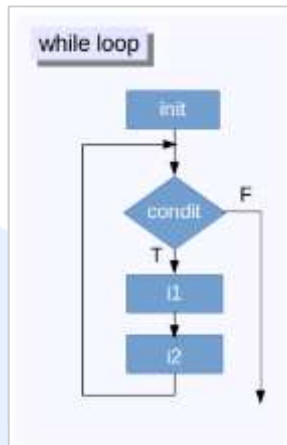
루프를 멈추는데 break 사용

```
> sum = 0
> for (i in 1:10) {
+   sum <- sum + i
+   if (sum>20){
+     # stop loop after sum>20
+     break
+   }
+   print(sum)
+ }
[1] 1
[1] 3
[1] 6
[1] 10
[1] 15
```


루프문 (while)

✓ while을 사용한 루프

➤ 1부터 5까지의 숫자 출력



y가 5보다 적을 때는
{expression}부분을 수행

```
# while loop  
# while (condition) {expression}  
y=0  
while(y <5){ print(y<-y+1) }
```



컨솔창

```
> y=0  
> while(y <5){ print(y<-y+1) }  
[1] 1  
[1] 2  
[1] 3  
[1] 4  
[1] 5
```