

3주차 사용자 정의 함수 및 제어문 -교육부-



Management Information System Of the Ubiquitous era

2.6.1 나만의 함수 만들기





2.6.1 함수 생성 및 함수의 특징



함 수 의 기 본 구 조

함수(function)란?

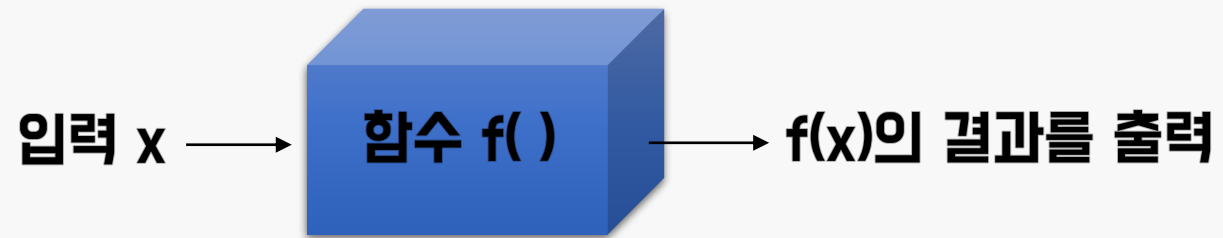
목적을 수행하기 위해 잘 정의된 독립적인 단위
입력을 받아 계산(결과)의 출력, 그래프 생성, 다른 함수의 호출 등)을 함.

```
function(입력항목,입력항목,...){
```

```
    함수에서 사용할 구문
```

```
    return(반환할객체)
```

```
}
```





2.6.1 함수 생성 및 함수의 특징



함 수 의 기 본 구 조

◆ 인수가 없는 함수의 예시 - 실행하면 1을 출력하는 함수

```
> myfunc01<- function(){  
+   return(1)  
+ }  
> myfunc01()  
[1] 1
```

- return(): 계산 결과를 반환

◆ 두 입력 항목을 더해 2를 곱하는 함수 생성

```
> my_fun <- function(i, j){  
+   k <- (i+j)*2  
+   return(k)  
+ }  
> # 생성된 객체의 형식 확인  
> class(my_fun)  
[1] "function"  
> # 함수 실행: (2+3)*2 = 10  
> my_fun(2, 3)  
[1] 10
```

- i,j: 인수로써 함수이름(i,j)를 입력 받게 된다.



2.6.1 함수 생성 및 함수의 특징



함 수 의 기 본 구 조

- ◆ 두 입력 항목을 더해 2를 곱하는 함수 - 객체에 값을 지정해 return을 수행하지 않고 바로 작성
- ◆ return 구문이 없을 경우 마지막 구문의 결과를 반환

```
> my_fun <- function(i, j){ (i+j)*2 }  
> my_fun(2, 3)  
[1] 10
```

- 간단한 함수일 경우 코드의 간결성을 위해 return을 생략 하기도함.
- 함수가 복잡해질 경우 코드의 이해를 돕기 위해 return문을 명시하는 것을 권장.



2.6.1 함수 생성 및 함수의 특징



함 수

연 습 문 제

1. 인수로 수치를 넣으면 입력한 수치를 제공한 값의 결과로 반환하는 함수 mypower01()을 정의하라.

```
> mypower01(3)           # 3의 제곱을 계산한다.  
[1] 9
```

2. 인수로 2개의 수치 x, y를 넣으면 x의 y승을 결과로 반환하는 함수 mypower02()를 정의하라.

```
> mypower02(3, 4)        # 3의 4제곱을 계산한다.  
[1] 81
```



2.6.1 함수 생성 및 함수의 특징



함 수



연 습 문 제

◆ 1번 문제 정답

```
> mypower01<-function(x){  
+   return(x^2)  
+ }  
> mypower01(3)  
[1] 9
```

◆ 2번 문제 정답

```
> mypower02<-function(x,y){  
+   return(x^y)  
+ }  
> mypower02(3,4)  
[1] 81
```



2.6.1 함수 생성 및 함수의 특징



함 수 의 기 본 구 조

◆ 함수에서의 모든 변경은 함수 내에서만 유효하다.

```
> input <- c(1, 2, 3)
> my_fun2 <- function(obj){
+   obj <- obj*2
+   input <- obj # 함수 안에서 계산된 값을 함수 바깥에 있는 input 객체에 지정
+   return(obj)
+ }
> my_fun2(input)
[1] 2 4 6
> input
[1] 1 2 3
```

- 함수를 실행 했으나 input이라는 객체의 값은 바뀌지 않음.

◆ 함수 실행 결과를 "<-"로 지정

```
> input <- my_fun2(input)
> input
[1] 2 4 6
```




2.6.1 함수 생성 및 함수의 특징



함 수 의 기 본 구 조

◆ "<<-"를 이용해 함수 밖의 객체를 직접 수정

```
> input <- c(1,2,3)
> my_fun2 <- function(obj){
+     obj <- obj*2
+     input <<- obj # "<<-" 연산자로 함수 바깥의 객체에 값 지정
+     return(obj)
+ }
> my_fun2(input)
[1] 2 4 6
> input
[1] 2 4 6
```

- 후에 코드를 확인하거나 추후 코딩을 진행 하는 과정에서 혼선을 줄 수 있으므로 권장하지 않음!!

2.6.2 제어문





2.6.2 제어문

2.6.2.1 if/else



기



본



구



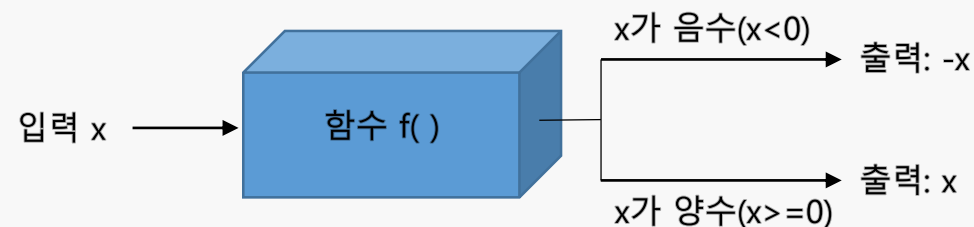
문

◆ 기본구문

- 특정 조건에서 처리를 다르게 하고 싶을 때 if문을 사용.
- 조건식이 false인 경우 else의 값이 출력

```
if(조건){  
    조건에 맞을 때 실행할 구문  
}  
else{  
    조건에 맞지 않을 때 실행할 구문  
}
```

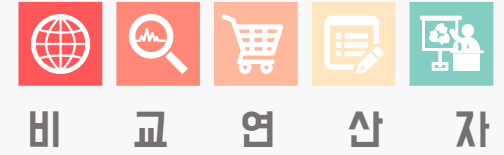
이 때, 조건은 logical로 TRUE나 FALSE가 되어야한다.





2.6.2 제어문

2.6.2.1 if/else



◆ 비교연산자 목록

기호	의미
==	같다.
!=	같지 않다.
>=	크거나 같다(이상).
>	크다
<=	작거나 같다(이하).
<	작다(미만).

◆ example

if (x == 1)	<처리 내용>	# x가 1과 같으면
if (x != 2.0)	<처리 내용>	# x가 2.0 이 아니면
if (x >= 3)	<처리 내용>	# x가 3 이상이면
if (x < 4.0)	<처리 내용>	# x가 4.0 미만이면



2.6.2 제어문

2.6.2.1 if/else



◆ if문을 포함한 함수 생성

```
> my_fun3 <- function(score){  
+   if(score >= 80){  
+       "합격"  
+   }else{  
+       "불합격"  
+   }  
+ }
```

- 80점 이상이면 합격, 아니면 불합격

◆ 함수 실행 결과

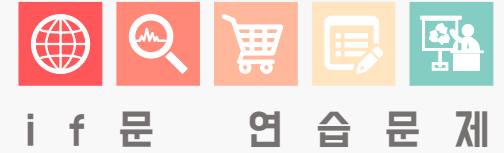
```
> # 70점일 때  
> 70>=80  
[1] FALSE  
> my_fun3(70)  
[1] "불합격"  
>  
> # 90점일 때  
> 90>80  
[1] TRUE  
> my_fun3(90)  
[1] "합격"
```

- 70점인 경우 false 이므로 불합격출력
- 90점인 경우 TRUE 이므로 합격 출력



2.6.2 제어문

2.6.2.1 if/else



◆ 1번 문제.

- 인수 x 가 1보다 클 때는 1을 출력하고 1보다 크지 않을 때는 0을 출력하는 함수 `myindex()`를 정의하라.

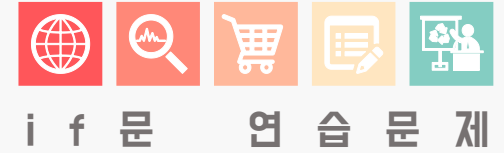
◆ 2번 문제.

- a, b 두 수치를 넣으면 입력한 수치의 차(무조건 양수)를 결과로 계산해서 돌려주는 함수 `mydistance()`를 정의하라.



2.6.2 제어문

2.6.2.1 if/else



i f 문 연 습 문 제

◆ 1번 문제.

- ◆ 1보다 클 때 1을 출력,
- ◆ 1보다 크지 않을 때 0 출력

```
> myindex<-function(x){  
+   if(x>1){  
+     return(1)  
+   }else{  
+     return(0)  
+   }  
+ }  
> myindex(2)  
[1] 1
```

◆ 2번문제.

- ◆ 두 수치의 차를 계산해서 돌려주는 함수

```
> mydistance<-function(a,b){  
+   if(a-b>=0){  
+     return(a-b)  
+   }else{  
+     return(b-a)  
+   }  
+ }  
> mydistance(3,2)  
[1] 1  
> mydistance(2,3)  
[1] 1
```



2.6.2 제어문

2.6.2.1 if/else



◆ ifelse 함수

- ifelse(조건, 조건에 맞을 때 실행할 구문, 조건에 맞지 않을 때 실행할 구문)

◆ 예시

```
> my_fun4 <- function(score){  
+ ifelse(score >= 80, "합격", "불합격")  
+ }  
> my_fun4(70)  
[1] "불합격"  
> my_fun4(90)  
[1] "합격"
```

- my_fun4 라는 함수 지정
- 인수가 80이상이면 합격, 그 외에 불합격 출력



2.6.2 제어문

2.6.2.1 if/else



i f e l s e 합 수

◆ 다중 ifelse

- ifelse(조건, 조건에 맞을 때 실행할 구문, ifelse(조건, 조건에 맞을 때 실행할 구문, 조건에 맞지 않을 때 실행할 구문))

◆ 예시

```
> my_fun4_1 <- function(score){  
+   ifelse(score >= 80, "고득점",  
+   ifelse(score >= 60 & score < 80, "평타", "망함"))  
+ }  
> my_fun4_1(70)  
[1] "평타"  
> my_fun4_1(59)  
[1] "망함"
```

- my_fun4 라는 함수 지정
- 인수가 80이상이면 "고득점", 60이상 80미만이면 "평타", 그 외에 "망함"을 출력
- Ifelse가 많아지면 괄호가 많아지므로 주의!!



2.6.2 제어문

2.6.2.1 if/else



i f e l s e 연 습 문 제

◆ 연습문제 - ifelse를 이용한 재범주화

iris 데이터에 Sepal.Length가 5미만인 경우 very short,
5이상 6미만인 경우 short,
6이상 7미만인 경우 long,
7이상인 경우 very long 으로 바꾸세요.
바뀐 Sepal.Length의 범주를 확인하세요.



2.6.2 제어문

2.6.2.1 if/else



i f e l s e 합 수

◆ 정답 코드

```
iris$Sepal.Length <- with(iris, ifelse(Sepal.Length < 5, "very short",  
                                     ifelse(Sepal.Length >= 5 & Sepal.Length < 6, "short",  
                                             ifelse(Sepal.Length >= 6 & Sepal.Length < 7, "long", "very long"))))  
table(iris$Sepal.Length)
```

◆ 범주 확인

```
> table(iris$Sepal.Length)
```

long	short	very long	very short
54	61	13	22

- with(data,...) : 같은 객체 이름 혹은 데이터 이름이 반복 될 때 사용
- with(iris,...)로 iris를 지정해줌으로써 뒤에 iris 생략. 코드가 간결해짐.



2.6.2 제어문

2.6.2.1 if/else



◆ else/if

- 만약 조건이 하나 이상인 경우 else if를 통해 조건을 추가할 수 있다. 여러 번 추가 가능
- else if (조건) { 조건 해당 시 수행할 구문 }

◆ 예시

```
> my_fun5 <- function(score){  
+   if(score >= 80){  
+     "합격"  
+   } else if(score >= 75){  
+     "대기"  
+   } else{  
+     "불합격"  
+   }  
+ }  
> my_fun5(70)  
[1] "불합격"  
> my_fun5(78)  
[1] "대기"  
> my_fun5(90)  
[1] "합격"
```

- 80점 이상은 if 절에서 처리
- 75점 이상인 경우 else if 절에서 처리
- 그 외 else 절에서 처리



2.6.2 제어문

2.6.2.1 if/else



◆ ifelse 함수를 통해 구현

```
> my_fun5_1<-function(score){  
+   ifelse(score>=80,"합격",  
+         ifelse(score>=75,"대기","불합격"))  
+ }  
> my_fun5_1(78)  
[1] "대기"
```



2.6.2 제어문 2.6.2.2 for



f



o



r

◆ for : 특정 구문을 반복적으로 수행

- `for(x in 시작인덱스:종료인덱스){`
 반복해서 처리할 구문
}

◆ 예시

```
> my_fun6 <- function(num){  
+   for(x in 1:10){  
+     print(num + x)  
+   }  
+ }  
> # 함수 실행  
> my_fun6(10)  
[1] 11  
[1] 12  
[1] 13  
[1] 14  
[1] 15  
[1] 16  
[1] 17  
[1] 18  
[1] 19  
[1] 20
```

- 함수 생성 - 입력받은 값에 1씩 더하면서 출력
- x값이 1, 2, 3, 4, 5 ... 8, 9, 10으로 변경되면서 수행됨



2.6.2 제어문 2.6.2.2 for



f



o



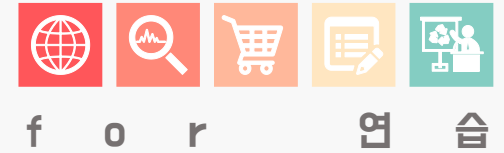
r

◆ start index와 end index는 꼭 1:10 이럴 필요는 없다.

```
> myloop2<- function() {  
+   a<- 0  
+   for (i in 3:7) {  
+     a<-a+1  
+   }  
+   return(a)  
+ }  
> myloop2()  
[1] 5
```



2.6.2 제어문 2.6.2.2 for



◆ for 문 연습

- ◆ 정수 n을 넣으면 1에서 n까지의 합을 돌려주는 함수 mysum()

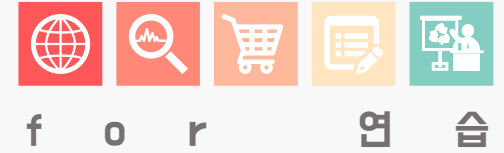
```
> mysum<- function(n) {  
+   i<- 0  
+   for (j in 1:n) {  
+     i<-i+j  
+   }  
+   return(i)  
+ }  
> mysum(10)  
[1] 55
```

◆ 헛갈릴 경우 print를 통해서 확인 가능

```
> mysum_1<- function(n) {  
+   i<- 0  
+   for (j in 1:n) {  
+     i<-i+j  
+     print(i)  
+   }  
+   return(i)  
+ }  
> mysum_1(10)  
[1] 1  
[1] 3  
[1] 6  
[1] 10  
[1] 15  
[1] 21  
[1] 28  
[1] 36  
[1] 45  
[1] 55  
[1] 55
```




2.6.2 제어문 2.6.2.2 for



◆ for 문 연습

◆ for문을 이용한 append (객체쌓기)

```
> b<-c()
> num<-10
> for (i in 1:10){
+   b<-c(b,num+i)
+ }
> b
[1] 11 12 13 14 15 16 17 18 19 20
```

◆ 구구단에서 3단을 for문을 사용하여 출력하기

```
> n<-3
> m<-1:9
> for(i in m){
+   times<- n * i
+   print(paste(n, "x", i, "=", times))
+ }
[1] "3 x 1 = 3"
[1] "3 x 2 = 6"
[1] "3 x 3 = 9"
[1] "3 x 4 = 12"
[1] "3 x 5 = 15"
[1] "3 x 6 = 18"
[1] "3 x 7 = 21"
[1] "3 x 8 = 24"
[1] "3 x 9 = 27"
```



2.6.2 제어문 2.6.2.2 for



f o r 연 습 문 제

◆ 1번 문제.

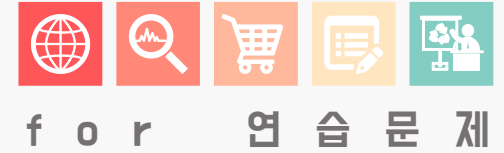
- 1에서 x까지의 짝수를 모두 더하는 함수 myeven()을 정의하라.
hint ($x \% y$: x를 y로 나눴을때 나머지 반환, **next**: 다음으로 넘어감)

◆ 2번 문제.

- 벡터 $x \leftarrow c(1,2,3,NA,4,5,NA)$ 를 할당하고, NA는 더하지 않고 다른 요소만을 더하는 함수 myplus()를 정의하라.
- **hint**: length, is.na() or !is.na()



2.6.2 제어문 2.6.2.2 for



f o r 연 습 문 제

◆ 1번문제.

- ◆ 1에서 x까지의 짝수를 모두 더하는 함수 myeven()을 정의하라

```
> myeven<-function(x){  
+   a<-0  
+   for(i in 1:x){  
+     if(i%%2==0){  
+       a<-a+i  
+     }else{  
+       next  
+     }  
+   }  
+   return(a)  
+ }  
> myeven(10)  
[1] 30
```

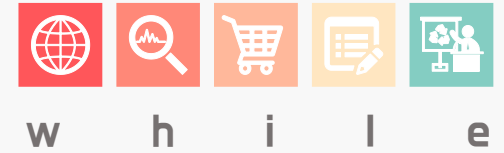
◆ 2번문제.

- ◆ 벡터 x<-c(1,2,3,NA,4,5,NA)를 할당하고, NA는 더하지 않고 다른 요소만을 더하는 함수 myplus()를 정의하라.

```
> x<-c(1,2,3,NA,4,5,NA)  
> myplus<-function(vec){  
+   d<-0  
+   for(i in 1:length(vec)){  
+     if(is.na(vec[i])){  
+       next  
+     }else{  
+       d<-d+vec[i]  
+     }  
+   }  
+   return(d)  
+ }  
> myplus(x)  
[1] 15
```



2.6.2 제어문 추가. while



◆ for : 어떤 특정 조건을 만족하는 한 계산을 계속 수행.

◆ for()루프를 이용하여 반복을 조절하는것이 어려울때 사용.

◆ while(condition) {
◆ <...반복할 식...>
◆ }

condition에 해당하는 조건문이 TRUE인 경우 작업 수행.
condition에 해당하는 조건문이 FALSE인 순간 작업 정지.

```
> x<-0
> y<-0
> while(y<=10){
+   x<-x+y
+   y<-y+1
+ }
> x ; y
[1] 55
[1] 11
```

- Y<=10 일 때 까지 수행하므로 x는 1~10까지 더하지만 최종 y는 11이 됨.
- 사실 전 잘 안써요.. 그냥 for문 쓰래요..

2.7 객체 조회/삭제/ 저장/불러오기





객체 다루기



후



우



.



.

각종 개체 관련 함수

함수	기능
ls()	객체 목록 조회
rm(객체, list=character(), ...)	객체 삭제
save(객체, ..., file="파일 명.rdata")	특정 객체를 파일로 저장
save.image("파일 명.rdata")	모든 객체를 파일로 저장
load("파일명.rdata")	파일로부터 객체 불러오기
setwd("경로")	작업 폴더 변경
getwd()	작업 폴더 확인

활용

```
> ls()
character(0)
> v_vector <- c(1, 2, 3, 4, 5)
> v_df <- data.frame(v_vector)
> ls()
[1] "v_df"      "v_vector"
> getwd()
[1] "C:/Users/dudtj/Desktop"
> setwd("C:/Users/dudtj/Desktop")
> getwd()
[1] "C:/Users/dudtj/Desktop"
> save(v_vector, v_df, file="save_test.rdata")
> rm(list=ls())
> load("save_test.rdata")
> v_vector
[1] 1 2 3 4 5
```

2.8 데이터 불러오기



데이터를 R 품에



데 이 터 불 러 오 기

◆ read.table(), read.csv()

```

> getwd()
[1] "C:/Users/dudtj/Desktop"
> ex_csv_df <- read.table(file="exam.csv", header=TRUE, sep=";", stringsAsFactors=FALSE)
> str(ex_csv_df)
'data.frame':   4 obs. of  3 variables:
 $ 이름: chr   "김영석" "이정민" "최지은" "고광민"
 $ 나이: int   24 24 23 23
 $ 성별: chr   "남" "남" "여" "남"
> ex_csv_df
  이름 나이 성별
1 김영석  24   남
2 이정민  24   남
3 최지은  23   여
4 고광민  23   남
> ex_csv_df<-read.csv("C:/Users/dudtj/Desktop/exam.csv")
> str(ex_csv_df)
'data.frame':   4 obs. of  3 variables:
 $ 이름: Factor w/ 4 levels "고광민","김영석",...: 2 3 4 1
 $ 나이: int   24 24 23 23
 $ 성별: Factor w/ 2 levels "남","여": 1 1 2 1

```

- read.table : txt 파일이나 csv 파일 등을 불러오기 가능
- read.csv : csv 파일 불러오기 가능, 구분자를 지정해줄 필요 없음
- Header : 첫 행이 변수명이면 TRUE
- sep : 구분자, 파일에 각 값이 어떻게 구분 되어있는지 지정 (csv파일은 ";" 으로 지정 되어있음)
- file 명에 경로를 지정해주지 않으면 지정 되어있는 경로에서 불러옴, 경로 직접 지정 가능



데이터를 R 품에



◆ write.table(), write.csv()

```
> write.table(ex_csv_df, file="test_file_save.csv", sep = ",", col.names=TRUE, row.names=FALSE, append=FALSE)
> write.csv(ex_csv_df, "test_file_save.csv", row.names=F)
> write.table(ex_csv_df, file="test_file_save.csv", sep=";", col.names=FALSE, row.names=FALSE, append=TRUE)
```

- write.table : 데이터를 txt 파일 혹은 csv 파일 등으로 저장 (ex_csv_df : 데이터명)
- write.csv : 데이터를 csv 파일로 저장
- col.names : 변수명 넣기
- row.names : 행 이름, 혹은 인덱스 넣기
- Append : 기존에 있는 데이터에 추가하는 것인지에 대한 여부
test_file_save.csv 라는 파일에 ex_csv_df 라는 데이터를 추가로 rbind 형식으로 덧붙인다.

2.9 패키지





튼튼한 패키지 친구들



패



키



지

◆ 2.9.1 패키지 설치 및 로드

- 패키지 설치 : `install.packages("패키지명")`
- 패키지 로드 : `library("패키지명")`
- 패키지 언로드 : `detach("package : 패키지명")`
- 패키지 제거 : `remove.packages("패키지명")`

패키지를 로드 하지 않으면 패키지에서 제공하는 함수나 help문을 사용할 수 없습니다!!

거의 끝!



튼튼한 패키지 친구들



설치는 미리 했어요!

◆ 패키지 오류 예시

```
> str_detect(c("hello bebe", "oh happy day"), "b")
Error in str_detect(c("hello bebe", "oh happy day"), "b") :
  could not find function "str_detect"
```

```
> library("stringr")
```

Warning message:

패키지 'stringr'는 R 버전 3.5.3에서 작성되었습니다

```
> str_detect(c("hello bebe", "oh happy day"), "b")
```

```
[1] TRUE FALSE
```

```
> detach("package:stringr")
```

```
> str_detect(c("hello bebe", "oh happy day"), "b")
```

```
Error in str_detect(c("hello bebe", "oh happy day"), "b") :
  could not find function "str_detect"
```

```
> remove.packages("stringr")
```

Removing package from 'C:/Users/dudtj/Documents/R/win-library/3.5'
(as 'lib' is unspecified)

```
> library("stringr")
```

```
Error in library("stringr") : 'stringr'이라고 불리는 패키지가 없습니다
```

```
> ? str_detect
```

```
No documentation for 'str_detect' in specified packages and libraries:
you could try '??str_detect'
```

- library가 load 되지 않아 오류발생

- 경고는 무시하셔도 됩니다.

- 패키지를 언로드 하여 다시 함수 사용 불가

- 패키지를 제거하여 로드 불가

- 패키지가 로드 되어있지 않으므로
help 사용불가

Thank you
Q&A