



# 4주차: 데이터 취득과 정제

**ChulSoo Park**

School of Computer Engineering & Information Technology

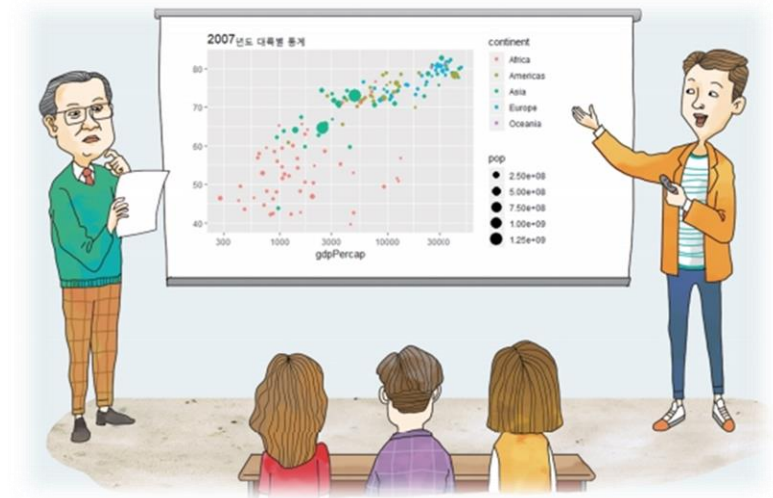
Korea National University of Transportation



# 04

## CHAPTER

# 데이터 취득과 정제



## CONTENTS

- 4.1 파일 일고 쓰기
- 4.2 데이터 정제를 위한 조건문과 반복문
- 4.3 사용자 정의 함수 : 원하는 기능 묶기
- 4.4 데이터 정제 예제 1 : 결측값 처리
- 4.5 데이터 정제 예제 2 : 이상값 처리
- 요약



## 4.1 파일 읽고 쓰기

- read.csv 함수: 일반 텍스트 파일 읽기 (c:/rdata/sudents.csv)

R: Data Input ▾

Find in Topic

```
read.table(file, header = FALSE, sep = "", quote = "\"",  
           dec = ".", numerals = c("allow.loss", "warn.loss",  
           row.names, col.names, as.is = !stringsAsFactors,  
           na.strings = "NA", colClasses = NA, nrow = -1,  
           skip = 0, check.names = TRUE, fill = !blank.lines.  
           strip.white = FALSE, blank.lines.skip = TRUE,  
           comment.char = "#",  
           allowEscapes = FALSE, flush = FALSE,  
           stringsAsFactors = default.stringsAsFactors(),  
           fileEncoding = "", encoding = "unknown", text, ski  
  
read.csv(file, header = TRUE, sep = ",", quote = "\"",  
         dec = ".", fill = TRUE, comment.char = "", ...)
```



## 4.1 파일 읽고 쓰기

- read.csv 함수: 일반 텍스트 파일 읽기 (c:/rdata/sudents.csv)

로컬 디스크 (C:) > rdata

이름	수정한 날짜	유형
students.csv	2021-03-14 오전 9:38	Microsoft Excel 싼표로 구분된 값 파일
students.txt	2021-03-14 오전 6:33	텍스트 문서
students1.txt	2021-03-14 오전 9:12	텍스트 문서
students1_old.txt	2021-02-14 오전 9:37	텍스트 문서
students2.csv	2021-02-14 오전 9:56	Microsoft Excel 싼표로 구분된 값 파일
students2.txt	2021-03-14 오전 9:31	텍스트 문서
students2_old.txt	2021-02-14 오전 9:50	텍스트 문서
students3.csv	2021-02-14 오후 8:20	Microsoft Excel 싼표로 구분된 값 파일
studentsxls.xlsx	2021-02-14 오전 9:59	Microsoft Excel 워크시트

	A	B	C	D	E
1	name	korean	english	math	
2	박철수	100	90	100	
3	김영희	90	100	80	
4	김영철	90	95	90	
5	손흥민	100	85	95	
6	류현진	85	100	100	
7					
8					




## 4.1 파일 읽고 쓰기

- read.csv 함수: 일반 텍스트 파일 읽기 (c:/rdata/sudents.csv)

```
students=read.csv("C:/rdata/students.csv")  
students  
str(students)
```

	A	B	C	D	E
1	name	korean	english	math	
2	박철수	100	90	100	
3	김영희	90	100	80	
4	김영철	90	95	90	
5	손흥민	100	85	95	
6	류현진	85	100	100	
7					
8					

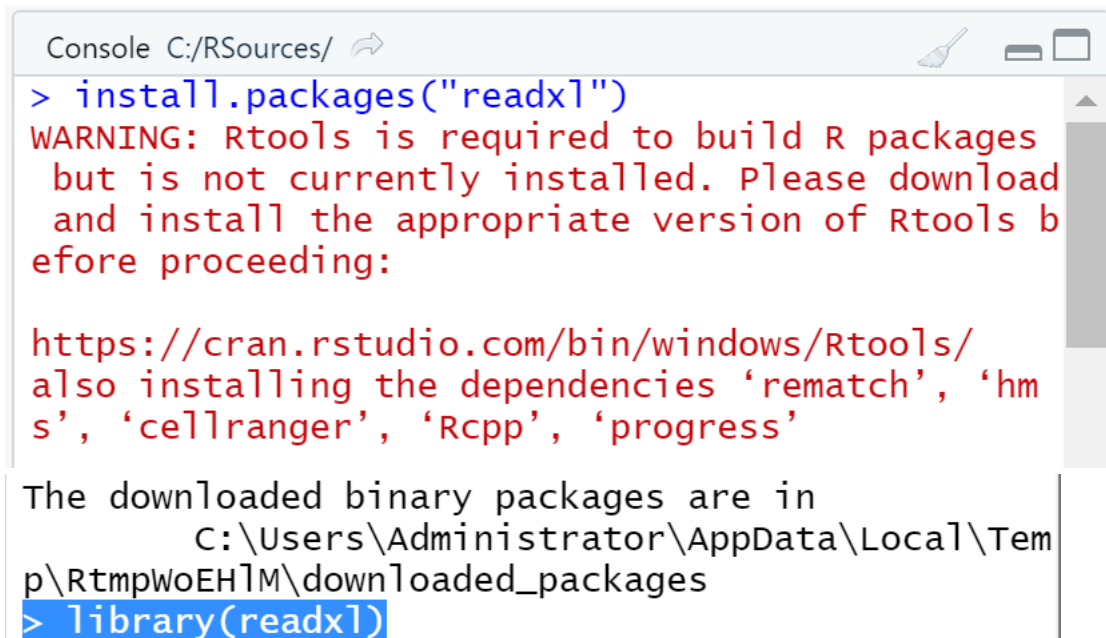
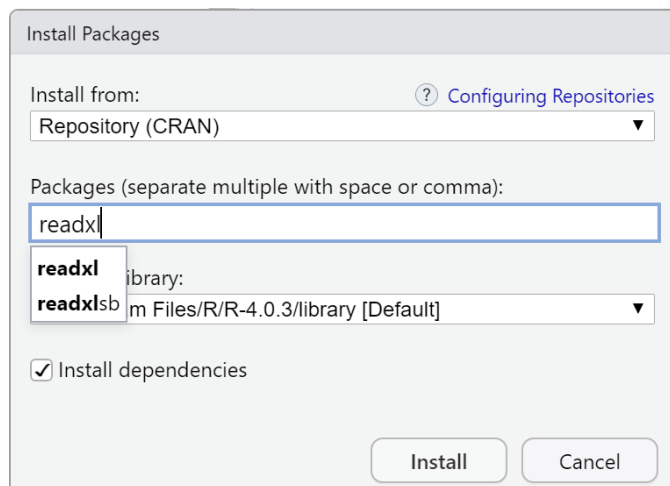
```
Console C:/Rsources/   
> students  
  name korean english math  
1 박철수    100      90   100  
2 김영희     90     100    80  
3 김영철     90      95    90  
4 손흥민    100      85    95  
5 류현진     85     100   100  
> str(students)  
'data.frame':  5 obs. of  4 variables:  
 $ name      : chr  "박철수" "김영희" "김영철" "손흥민" ...  
 $ korean    : int   100  90  90 100 85  
 $ english   : int   90 100 95 85 100  
 $ math      : int  100 80 90 95 100
```



## 4.1 파일 읽고 쓰기

- read.xls, read.xlsx 함수: 일반 텍스트 파일 읽기 (c:/rdata/sudents.xlsx)

RStudio>Tools>Install Packages



## 4.1 파일 읽고 쓰기

- read.xls, read.xlsx 함수: 일반 텍스트 파일 읽기 (c:/rdata/sudents.xlsx)

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

4chapter.R\* x R data sets x

Source on Save Run Source

```

18 students=read.table("C:/rdata/students1.txt",sep=" ",header=T,as.is=T,na.strings="NA")
19 students=read.table("C:/rdata/students2.txt",sep=" ",header=T)
20
21 students=read.csv("C:/rdata/students.csv")
22 library(readxl)
23 students=read_excel("C:/rdata/students.xlsx")
24 students
25 str(students)
26
27 readxl
28
29

```

readxl: Read Excel Files

Import excel files into R. Supports '.xls' via the embedded 'libxls' C library <<https://github.com/libxls/libxls>> and '.xlsx' via the embedded 'RapidXML' C++ library <<http://rapidxml.sourceforge.net>>. Works on Windows, Mac and Linux without external dependencies.

Press F1 for additional help

Environment Histor

R Global Environ

students 5 ob

tmp1 338

y\_c\_pop 60 c

Files Plots Packa

Data Input Fir

lines and variable

sage

```

read.table(file
dec
row.
na.s
skip
stri
comm
allc
stri
fir

```

name

```

<chr>
1 박철수
2 김영철
3 김영철 90 95 90
4 손흥민 100 85 95
5 류현진 85 100 100
> str(students)
tibble [5 x 4] (s3: tbl_df/tbl/data.frame)
 $ name : chr [1:5] "박철수" "김영철" "김영철" "손흥민" ...
 $ korean : num [1:5] 100 90 90 100 85
 $ english: num [1:5] 90 100 95 85 100
 $ math : num [1:5] 100 80 90 95 100

```



## 4.1 파일 읽고 쓰기

- read.xls, read.xlsx 함수: 일반 텍스트 파일 읽기 (c:/rdata/sudents.xlsx)

```
Console C:/RSources/
$ english: int 100 80 90 95 100
$ math   : int 100 80 90 95 100
> students=readxl("C:/rdata/sudents.xlsx")
Error in readxl("C:/rdata/sudents.xlsx") :
  함수 "readxl"를 찾을 수 없습니다
> library(readxl)
> students=readxl("C:/rdata/sudents.xlsx")
Error in readxl("C:/rdata/sudents.xlsx") :
  함수 "readxl"를 찾을 수 없습니다
> search()
[1] ".GlobalEnv"          "package:readxl"      "tools:rstudio"       "package:stats"
[5] "package:graphics"    "package:grDevices"   "package:utils"        "package:datasets"
[9] "package:methods"     "AutoLoads"           "package:base"
> students=read_xlsx("C:/rdata/sudents.xlsx")
> students
# A tibble: 5 x 4
  name      korean english  math
  <chr>    <dbl>    <dbl> <dbl>
1 박철수    100      90    100
2 김영희     90    100     80
3 김영철     90     95     90
4 손흥민    100     85     95
5 류현진     85    100    100
> str(students)
tibble [5 x 4] (S3: tbl_df/tbl/data.frame)
 $ name      : chr [1:5] "박철수" "김영희" "김영철" "손흥민" ...
 $ korean    : num [1:5] 100 90 90 100 85
 $ english   : num [1:5] 90 100 95 85 100
 $ math      : num [1:5] 100 80 90 95 100
```





## 4.1 파일 읽고 쓰기

### ② 파일 쓰기

- write.table 함수: 일반 텍스트 파일로 저장할 때 사용 : .txt
- Write.csv 함수 : csv 파일로 저장 : .csv

Console C:/RSources/ ↗

> ?write.table

R: Data Output ▾ Find in Topic

#### Usage

```
write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ",  
            eol = "\n", na = "NA", dec = ".", row.names = TRUE,  
            col.names = TRUE, qmethod = c("escape", "double"),  
            fileEncoding = "")
```

```
write.csv(...)  
write.csv2(...)
```



## 4.1 파일 읽고 쓰기

- write.table 함수: 일반 텍스트 파일로 저장할 때 사용 : .txt

```
students=read.table("C:/rdata/students.txt",header=T)
```

```
write.table(students, file="c:/rdata/output.txt")
```

```
write.table(students, file="c:/rdata/output1.txt",quote=F)
```

```
write.table(students, file="c:/rdata/output2.txt",quote=F,row.names = FALSE)
```

output.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
"name" "korean" "english" "math"  
"1" "송민준" 100 90 100  
"2" "한태호" 90 100 80  
"3" "김남중" 90 95 90  
"4" "박재우" 100 85 95  
"5" "김연재" 85 100 100
```

output1.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
name korean english math  
1 송민준 100 90 100  
2 한태호 90 100 80  
3 김남중 90 95 90  
4 박재우 100 85 95  
5 김연재 85 100 100
```

output2.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

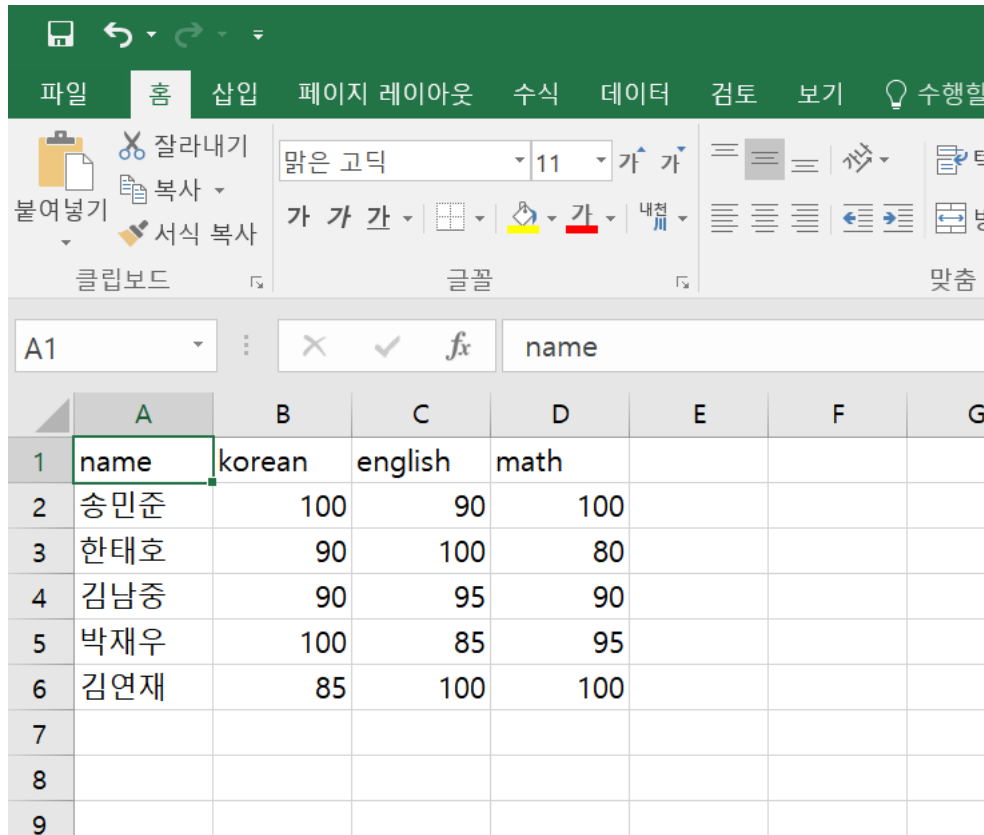
```
name korean english math  
송민준 100 90 100  
한태호 90 100 80  
김남중 90 95 90  
박재우 100 85 95  
김연재 85 100 100
```



## 4.1 파일 읽고 쓰기

- write.csv 함수: 일반 텍스트 파일로 저장할 때 사용 : .csv

```
students=read.table("C:/rdata/students.txt",header=T)
write.csv(students, file="c:/rdata/output.csv",quote=F,row.names = FALSE)
```



	A	B	C	D	E	F	G
1	name	korean	english	math			
2	송민준	100	90	100			
3	한태호	90	100	80			
4	김남중	90	95	90			
5	박재우	100	85	95			
6	김연재	85	100	100			
7							
8							
9							



## 4.2 데이터 정제를 위한 조건문과 반복문

### ① 조건문





- 데이터 정제를 위해 특정 조건에 맞는 값을 찾아내거나 일부 구간의 값을 추출하여 연산하는 등 다양한 목적에 맞게 작업할 수 있다.
- R에서 제공하는 조건 탐색 기능을 살펴보고, 조건문과 반복문 사용법에 대해 학습해 보자.
- 조건문 형식

조건에 맞는 요소를 추출하는 방법	형식
[ ]에 행/열 조건 명시	변수명 [행 조건식, 열 조건식]
If 문 활용 (if/ else if/else)	If(조건식) 표현식
Ifelse 문 활용	Ifelse(조건식, 참인 경우 반환값, 거짓인 경우 반환값)



## 4.2 데이터 정제를 위한 조건문과 반복문

- [ ]에 행/열 조건 명시
  - 벡터의 경우

```
Console C:/Rsources/      
> test=c(15,20,25,30,NA,40,45)  
> test[test<40]  
[1] 15 20 25 30 NA  
> test[test<40&!is.na(test)]  
[1] 15 20 25 30  
> test[test%%3!=0&!is.na(test)]  
[1] 20 25 40  
> test[is.na(test)]  
[1] NA  
> test[!is.na(test)]  
[1] 15 20 25 30 40 45
```



## 4.2 데이터 정제를 위한 조건문과 반복문

### ■ [ ]에 행/열 조건 명시

- 데이터 프레임의 경우

```
Console C:/RSources/
> characters=data.frame(name=c("길동","춘
향","철수"), age=c(30,16,21), gender=factor(c
("M","F","M")))
> characters
  name age gender
1 길동  30      M
2 춘향  16      F
3 철수  21      M
> characters[characters$gender=="F",]
  name age gender
2 춘향  16      F
> characters[characters$gender=="M",]
  name age gender
1 길동  30      M
3 철수  21      M
> characters[characters$gender=="M"&character
s$age<30,]
  name age gender
3 철수  21      M
```



## 4.2 데이터 정제를 위한 조건문과 반복문

### ■ if문 사용 (if, else if, else)

- 두 가지 조건 분기가 필요한 경우

Console C:/RSources/ ➔

```
> x=5
> if (x %% 2 == 0){
+   print(paste("x=",x,"는 짝수")) # 조건식이 참일 때 수행
+ } else {
+   print(paste("x=",x,"는 홀수")) # 조건식이 거짓일 때 수행
+ }
[1] "x= 5 는 홀수"
```

```
> x=4
> if (x %% 2 == 0){
+   print(paste("x=",x,"는 짝수")) # 조건식이 참일 때 수행
+ } else {
+   print(paste("x=",x,"는 홀수")) # 조건식이 거짓일 때 수행
+ }
[1] "x= 4 는 짝수"
```



## 4.2 데이터 정제를 위한 조건문과 반복문

- 세 가지 조건 분기가 필요한 경우

```
Console C:/Rsources/ ↗
[1] "x는 '적수'"
> x=-1
> if (x>0) {
+   print('x is a positive value.') # x가 0보다 크면 출력
+ } else if(x==0) {
+   print(' x is zero.') # x가 0이면 출력
+ } else {
+   print('x is negativ)e vlaue.') # 위의 모든조건을 만족하지 못하면 출력
+ }
[1] "x is negativ)e vlaue."
> x=0
> if (x>0) {
+   print('x is a positive value.') # x가 0보다 크면 출력
+ } else if(x==0) {
+   print(' x is zero.') # x가 0이면 출력
+ } else {
+   print('x is negativ)e vlaue.') # 위의 모든조건을 만족하지 못하면 출력
+ }
[1] " x is zero."
> x=11
> if (x>0) {
+   print('x is a positive value.') # x가 0보다 크면 출력
+ } else if(x==0) {
+   print(' x is zero.') # x가 0이면 출력
+ } else {
+   print('x is negativ)e vlaue.') # 위의 모든조건을 만족하지 못하면 출력
+ }
[1] "x is a positive value."
```










## 4.2 데이터 정제를 위한 조건문과 반복문

### ■ ifelse문 사용

- if/else 문을 합쳐놓은 형태
- 사용법: ifelse(조건식, 조건식이 참인 경우 반환값, 조건식이 거짓인 경우 반환값)

```
Console C:/RSources/      
> x=c(-5:5)  
> x  
[1] -5 -4 -3 -2 -1 0 1 2 3 4 5  
> ifelse(x>0,x+10,x)  
[1] -5 -4 -3 -2 -1 0 11 12 13 14 15
```

```
Console C:/RSources/   
> x=sample(1:100,10)  
> x  
[1] 53 89 52 67 81 55 91 99 22 73  
> x=sample(1:100,10)  
> x  
[1] 91 75 11 51 73 90 31 29 83 20  
> ifelse(x<50,x+20,x)  
[1] 91 75 31 51 73 90 51 49 83 40
```



## 4.2 데이터 정제를 위한 조건문과 반복문

- 예) 파일로부터 데이터를 읽어 들인 후 조건문 처리
  - 점수가 61~100점 이외의 값이 입력된 경우 NA로 처리하는 프로그램

Console C:/RSources/

```
> students=read.csv("c:/rdata/studentcsv3.csv")
> students
```

	name	korean	english	math
1	강서준	100	90	100
2	김도형	90	120	80
3	박정원	90	95	90
4	이상훈	100	85	10
5	최건우	85	100	100

```
> students[,2]=ifelse(students[,2]>=60&students[,2]<=100,students[,2],NA)
> students[,3]=ifelse(students[,3]>=60&students[,3]<=100,students[,3],NA)
> students[,4]=ifelse(students[,4]>=60&students[,4]<=100,students[,4],NA)
> students
```

	name	korean	english	math
1	강서준	100	90	100
2	김도형	90	NA	80
3	박정원	90	95	90
4	이상훈	100	85	NA
5	최건우	85	100	100

## 4.2 데이터 정제를 위한 조건문과 반복문

### ② 반복문

- 데이터 검토 시 반복적으로 값을 변경하면서 사용해야 하는 경우가 존재한다. 예를 들어, 데이터 프레임의 0번 행부터 10번 행까지 비교하는 등...
- R에서 제공하는 반복문은 repeat, while, for 문이 있다.

#### ■ 반복문 형식

반복문	의미
Repeat { 반복 수행할 문장 }	블록 안의 문장을 반복해서 수행한다
While(조건식) { 조건이 참일 때 수행할 문장 }	조건식이 참일 때 블록 안의 문장을 수행한다.
For {변수 in 데이터 { 반복 수행할 문장 }}	데이터의 각 요소를 변수에 할당하면서 각각에 대해 블록 안의 문장을 수행한다.



## 4.2 데이터 정제를 위한 조건문과 반복문

### ■ repeat 문 이용

- 1부터 5까지 수를 1씩 증가시키기

Console C:/RSources/ ➔

```
> # repeat 문 사용 사례
> i=1
> repeat {
+   if(i>5) {           # i가 5를 넘으면 중단
+       break
+   } else {
+       print(i)
+       i = i +1
+   }
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
└
```



## 4.2 데이터 정제를 위한 조건문과 반복문

### ■ while 문 이용

- 1부터 5까지 수를 1씩 증가시키기
- 구구단 2단 만들기

```
Console C:/Rsources/ ➤  
> # while 문 사용 사례  
> i=1  
> while (i<=5) {  
+   print(i)  
+   i = i +1  
+ }  
[1] 1  
[1] 2  
[1] 3  
[1] 4  
[1] 5
```

```
Console C:/Rsources/ ➤  
> # while 문 사용 사례(구구단 2단 만들기)  
> i=1  
> while(i<10) {  
+   print(paste(2, "x", i, "=", 2*i))  
+   i=i+1  
+ }  
[1] "2 x 1 = 2"  
[1] "2 x 2 = 4"  
[1] "2 x 3 = 6"  
[1] "2 x 4 = 8"  
[1] "2 x 5 = 10"  
[1] "2 x 6 = 12"  
[1] "2 x 7 = 14"  
[1] "2 x 8 = 16"  
[1] "2 x 9 = 18"
```



## 4.2 데이터 정제를 위한 조건문과 반복문

### ■ for 문 이용

- 1부터 5까지 수를 1씩 증가시키기

```
> # for 문을 이용하여 1부터 5까지 증가시키기
> for(i in 1:5){
+   print(i)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

- 구구단 2~9 단 만들기

7:7 → 2:9

```
Console C:/RSources/
> # for 문을 이용하여 구구단 2~9단 만들기
> for (i in 7:7) {
+   for(j in 1:9) {
+     print(paste(i, "x", j, "=", i*j))
+   }
+ }
[1] "7 x 1 = 7"
[1] "7 x 2 = 14"
[1] "7 x 3 = 21"
[1] "7 x 4 = 28"
[1] "7 x 5 = 35"
[1] "7 x 6 = 42"
[1] "7 x 7 = 49"
[1] "7 x 8 = 56"
[1] "7 x 9 = 63"
```



## 4.2 데이터 정제를 위한 조건문과 반복문

- 예) 조건문과 반복문을 활용하여 특정 범위 내에서 조건에 맞는 값 찾기

1~20까지에서 3의 배수

```
Console C:/RSources/ ➤  
> for(i in 1:20) {  
+   if(i%%3==0) {  
+     print(i)  
+   }  
+ }  
[1] 3  
[1] 6  
[1] 9  
[1] 12  
[1] 15  
[1] 18
```

2~15까지에서 소수 출력

```
Console C:/RSources/ ➤  
> for(i in 2:15) {  
+   c=0  
+   for(j in 2:i) {  
+     if(i%%j==0) {  
+       c = c + 1  
+     }  
+   }  
+   if(c==1) {print(i)}  
+ }  
[1] 2  
[1] 3  
[1] 5  
[1] 7  
[1] 11  
[1] 13
```

소수 : 1보다 큰 자연수 중 소수가 아닌 것은 합성수라고 한다. 1과 그 수 자신 이외의 자연수로는 나눌 수 없는 자연수로 정의하기도 한다.



## 4.2 데이터 정제를 위한 조건문과 반복문

- 예) 조건문과 반복문을 활용하여 특정 범위 내에서 조건에 맞는 값 찾기

Console C:/Rsources/ ↗

```
> students=read.csv("c:/rdata/studentscsv3.csv")
> students
  name korean english math
1 강서준    100     90   100
2 김도형     90    120    80
3 박정원     90     95    90
4 이상훈    100     85    10
5 최건우     85    100   100
> for(i in 2:4) {
+ students[,i]=ifelse(students[,i]>=60&students[,i]<=100,students[,i],NA)
+ }
> students
  name korean english math
1 강서준    100     90   100
2 김도형     90    NA    80
3 박정원     90     95    90
4 이상훈    100     85    NA
5 최건우     85    100   100
```





## 4.3 사용자의 함수 : 반복문 원하는 기능 묶기

### ■ 함수

- 입력과 출력간의 관계식을 함수라고 할 수 있다.
- 사용자의 목적에 맞는 다양한 함수를 만들어 보자.

### ■ 사용자 정의 함수의 구조

```
함수명 = function(전달자1, 전달자2, 전달자3, ...) {  
    함수 동작 시 수행 프로그램  
    return(반환값)  
}
```



## 4.3 사용자의 함수 :반 복문 원하는 기능 묶기

- 예) 곱셈(multiplication)을 구하는 함수

Console C:/RSources/ ↗

```
> # 함수(function) 구현
> mult=function(i,j){
+   ij = i * j
+   return(ij)
+ }
>
> mult(100,88)
[1] 8800
> mult(12345,54267)
[1] 669926115
> mult(10,100)
[1] 1000
```



# Thank you

