

# R 제어문

# 조건문

- 조건문은 특정한 조건을 만족할 때에만 명령어를 실행하여 사용함
- 조건문의 가장 대표적인 if 형태
  - `if(조건){실행문}`
  - `if(조건){실행문1}else{실행문2}`
  - `if(조건1){실행문1}else if(조건2){실행문2}else{실행문3}`

# if

- if(조건){실행문} 구조
  - 지정된 조건을 만족할 때에만 실행문을 실행하는 구조
  - x가 5보다 크면 "x is large number."라는 문자열을 출력하는 명령문

```
x = 10
if(x > 5){
    print("x is large number.")
}
```

- x가 10으로 5보다 크기 때문에 조건을 만족하여 "x is large number."를 출력

# if ~ else

- if(조건){실행문1}else{실행문2} 구조
  - 지정된 조건을 만족할 때에는 실행문1, 조건을 만족하지 않을 때에는 실행문2를 실행하는 구조
  - x가 5보다 크면 "x is large number."를 출력하고, x가 5보다 크지 않으면 "x is not large number"를 출력하라는 명령

```
x = 4 if(x > 5){  
    print("x is large number.")  
} else {  
    print("x is not large number.")  
}
```

- x가 4로서 5보다 작기 때문에 "x is not large number."를 출력

# if ~ else if ~ else

- if(조건){실행문1}else if(조건2){실행문2}else{실행문3} 구조
  - 지정된 첫 번째 조건을 만족하면 실행문1, 두 번째 조건을 만족하면 실행문2, 첫 번째와 두 번째 조건을 모두 만족하지 않으면 실행문3을 실행하는 구조

- x가 10보다 크면 "x is large number."를 출력하고, x가 10보다 크지 않고 5보다 크면 "x is medium number"를 출력하고, x가 5보다 작으면 "x is small number."를 출력하라는 명령어

x = 20

```
if(x > 10){  
    print("x is large number.")  
}else if(x > 5){  
    print("x is medium number.")  
}else{  
    print("x is small number.")  
}
```

- x가 20으로 첫 번째 조건인 10보다 크기 때문에
  - "x is large number."를 출력

# ifelse()

- ifelse(조건문, 실행1, 실행2) 구조
  - 조건에 맞으면 실행1을 실행하고, 조건에 맞지 않으면 실행2를 실행하는 구조
  - ifelse() 함수는 주로 새로운 변수를 생성할 때에 많이 사용
    - 예 : diamonds의 price가 5,000달러 미만과 5,000달러 이상으로 구분하는 price.group을 생성하고자 하면 ifelse() 함수를 사용함
  - diamonds\$price.group = ifelse(diamonds\$price < 5000, "5000달러 미만", "5000달러 이상")

# ifelse()

- ifelse(조건문, 실행1, 실행2) 구조
  - 두 개의 값을 가지는 새로운 변수를 생성할 때 사용하지만 세 개 이상의 값을 가지는 새로운 변수를 생성할 때도 사용할 수 있음
    - 대신에 중첩으로 사용하면 됨
  - 예 : price가 5,000달러 미만인 것과 5,000달러 이상 ~ 10,000달러 미만, 10,000달러 이상인 세 개의 값을 가지는 price.group2를 생성
    - `diamonds$price.group2 = ifelse(diamonds$price < 5000, "5000달러 미만", ifelse(diamonds$price < 10000, "5000달러 이상 ~ 10000달러 미만", "10000달러 이상"))`

# 반복문

- 반복문은 데이터를 분석하는 과정에서 동일한 일을 반복적으로 하거나 비슷한 일을 반복적으로 하게 되면 반복문을 사용하여 짧은 라인의 명령어로 동일한 일을 처리할 수 있음 반복문 중에서 for가 대표적임
- for는 for(조건문){실행문} 구조로 되어있어 조건문을 만족할 때까지 실행문을 여러 번 실행함



# for

- for(조건문){실행문} 구조
  - "Hello, world!"라는 문자열을 10번 출력하는 방법
    - print("Hello, world!")
    - print("Hello, world!")
    - print("Hello, world!")
    - print("Hello, world!")
    - print("Hello, world!")
    - print("Hello, world!")
    - print("Hello, world!")
    - print("Hello, world!")
    - print("Hello, world!")
    - print("Hello, world!")
    - 10개의 명령어를 작성하거나 동일한 명령어를 10번 실행해야 함

# for

- for(조건문){실행문} 구조

```
for(i in 1:10){  
    print("Hello, world!")  
}
```

- 위의 **3줄**의 명령어를 이용하여 **"Hello, world!"**라는 문장을 **10번** 출력

# for

- for(조건문){실행문} 구조

- "Hello, 1", "Hello, 2", "Hello, 3", "Hello, 4", "Hello, 5", "Hello, 6", "Hello, 7", "Hello, 8", "Hello, 9", "Hello, 10"을 출력하는 방법
  - 출력되는 내용을 보면 숫자인 부분이 변경되는 것을 볼 수 있음

```
for(i in 1:10){  
  cat("Hello,", i, " ")  
}
```

- cat() 함수는 문자열을 출력하는 함수로 함수에 들어오는 여러 개의 문자를 하나로 묶어서 하나의 문자열을 출력함
- 은 하나의 문자열을 출력하고 커서의 위치는 다음 줄로 이동하라는 뜻
- for도 if처럼 중첩해서 사용할 수 있음