**1. 조건문**

**1.1 if 조건문**

**if** (조건) {

(조건이 True일 때 실행될) **문장 또는 명령어**

} **else** {

(조건이 False일 때 실행될) **문장 또는 명령어**

}

() : 소괄호 / {} : 중괄호 / [] : 대괄호

**1.2 if 조건문 (if.. else if.. else 순서)**

if (조건1) {

('조건1'이 True일 때 실행될) 문장 또는 명령어

} else if {

('조건1'이 아니고 '조건2'일 때 실행될 문장 또는 명령어)

} else {

('조건1'도 '조건2'도 아닐 때 실행될 문장 또는 명령어)

}

**1.3 ifelse 조건문**

if (조건, 참, 거짓)

예)

x < - c(1,2,3,4)

y < - c(4,1,6,7)

ifelse(x<y, x, y)

결과값 : [1] 1, 1, 3, 4

\*\* Keep Point

**ifelse는 중복사용도 가능합니다.**

<참고좌표>

나노3\_day (59~62행 참고)

<https://data-make.tistory.com/43>

**2. 루프문**

: 작업을 반복하는 R의 방법으로 시뮬레이션 프로그래밍에 유용한 도구입니다.

**2.1 Expand.Grid**

: n개의 벡터에 있는 요소의 모든 조합을 작성

: 가능한 모든 팩터 레벨의 조합을 만들어준다.

수식 : expand.grid(x,y)

* 주사위의 기대값은 각 눈의 값에 각 확률을 곱한 값의 합이다.

**2.2 FOR문**

for(변수 in 반복횟수) {실행문}

FOR(i, in 1:5) {print(rep(i, i))}

* FOR문 실행문에서 print(value)와 value의 차이

for (value in c("My", "second", "for", "loop")) { print(value) }

value의 결과값을 출력한다.

* 두 문장의 차이점

for (word in c("My", "second", "for", "loop")) { print(word) }

임시적으로 값을 출력한다.

**2.3 WHILE문**

while(조건){실행문}

: 조건이 참(**조건에 도달했을 때**)인 경우 계속 실행문을 반복

* print() : 중간에 체크할 때 사용
* return() : 저장 기능 / 실제값을 사용

**2.4 REPEAT문**

repeat {반복해서 수행할 문장}

repeat문 : 다음것까지 출력을 해준다.

시작하면서 조건이 만족됨.

while문 : 그 전까지 출력해준다.

위에서부터 조건을 만족시킨 상태에서 내려옴

**3. APPLY 문**

: 행 또는 열 단위의 연산을 쉽게 할 수 있도록 지원하는 함수이다.

1이 **행 단위 연산**, 2가 **열 단위 연산**

FOR 문을 조금 더 쉽게 사용하기 위한 문

apply(iris[,1:4], 1, mean) : 행 별로 평균을 내라는 의미

**3.1 lapply**

: list + apply을 의미하는 함수. 실행결과가 list 형태로 출력

3.2 **sapply**

: lapply 함수에서 사용자 편의성을 고려한 함수. 실행결과가 Vector 형태로 출력

**3.3 tapply**

: 요인(factor)를 기준으로 해서, 그룹별로 나누어서 통계 분석을 하고자 할 때 사용하는 함수

tapply(벡터, 그룹, 연산)

**SAMPLE 함수**

sample(x, size, replace = FALSE, prob = NULL)

sample(1:10, 3) #1부터 10사이 값에서 무작위로 3개 추출 [1] 1 5 3