R 프로그래밍

(10주차)

2016. 05. 07(토)

장운호

(ADP 002-0004)

목차

※ 지난 주 복습 및 과제 리뷰

Ⅰ. 날짜형 데이터 처리

Ⅱ. Get & Assign

Ⅲ. 데이터 조인(Join)

※ 데이터의 종류

데이터는 가독성과 구조(Schema) 및 계산 난이도(Calculability) 등을 기준으로 구분이 가능함.

[데이터 가독성 기준 구분]

텍스트 (Text) 데이터	• CSV • TSV • HTML • XML • SVG
바이너리 (Binary) 데이터	• docx • xlsx • pptx • psd • RData

[데이터 구조/계산 난이도 기준 구분]

정형 데이터	• 구축DB • 엑셀 등	구조(Schema)를 가지고 있으며, 바로 계산이 가능함. (Calculable)
반정형 데이터	• 로그(log) • XML • HTML 등	구조를 가지고 있으나, 약간의 변환을 거쳐야 계산이 가능함
비정형 데이터	• VoC • SNS 등	구조를 가지고 있지 않아서, 상당한 전처리 과정을 거쳐야 계산이 가능함

Ⅰ. 날짜형 데이터 처리

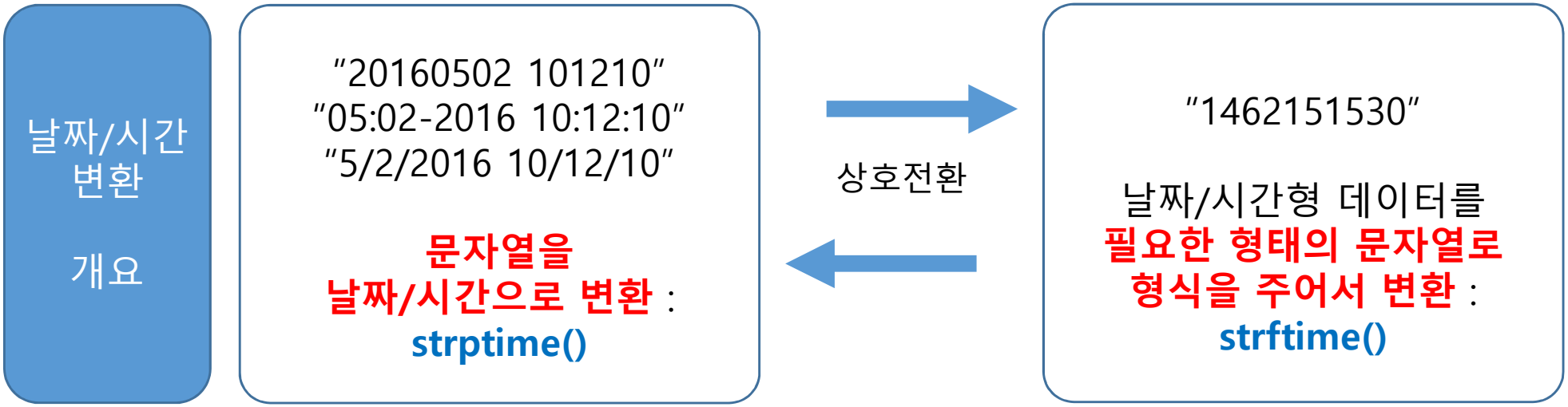
1. 날짜형 데이터의 타입 구분

1) POSIXct : 1970년 1월 1일 0시 0분 0초 부터의 경과 시간을 초단위로 세어서 10자리 숫자로 해당 시간을 표시함.

※ Portable Operating System Interface (X) Calendar Time

2) POSIXlt : 시, 분, 초, 요일, 매월초하루부터의 경과 날짜수, 매년 초하루부터의 경과 날짜수, 1900년 이후 경과된 년수, 썸머타임여부, 등을 timezone을 적용한 list 데이터 형태(mode)로 변환하고, 날짜형으로 해당시간을 표시함.

※ Portable Operating System Interface (X) Local Time



주) http://biostat.mc.vanderbilt.edu/wiki/pub/Main/ColeBeck/datestimes.pdf

2. 날짜형 데이터 처리

수집된 텍스트내의 날짜 표현 문자열을 날짜/시간형 데이터로 형변환이 가능함.

- 포맷 리터럴(literal) 지정을 통해 원하는 형태 즉, 주/월/시간대 등으로 변환하여 활용이 가능함.

Strptime 내부구성 (POSIXlt)

$sec	$min	$hour	$mday	$mon	$year
[1] 37	[1] 26	[1] 16	[1] 17	[1] 3[1)]	[1] 115

$wday	$yday	$isdst	$zone	$gmtoff
[1] 5[2)]	[1] 106	[1] 0	[1] "KST"	[1] NA

주: 1) 1월(0), 2월(1), 3월(2), 4월(3), 5월(4), 6월(5), 7월(6), … , 11월(10), 12월(11)
2) 일(0), 월(1), 화(2), 수(3), 목(4), 금(5), 토(6)

날짜형 포맷 지정 리터럴

리터럴	의미
%Y	연도를 4자리 숫자로 표시
%m	월을 2자리 이하 숫자로 표시
%d	날짜를 1부터 31의 숫자로 표시
%H	시간을 0부터 23의 숫자로 표시
%M	분을 0부터 59의 숫자로 표시
%S	초를 0부터 59의 숫자로 표시

리터럴	의미
%j	당해년도 몇번째 날짜(1~366)로 표시
%W	월요일 기준 당해년도 주차(00~56) 표시
%w	요일을 정수(0~6, 일요일 0)로 표시
%U	일요일 기준 당해년도 주차(00~56) 표시
%u	요일을 정수(1~7, 일요일 1)로 표시
%p	해당 타임존에 맞는 오전/오후 표시

3. 타임존 지정

날짜를 수치데이터로 바꾸어 프로그램적으로 처리하기 위해서는
시간카운트의 origin과 time zone을 별도의 Argument로 지정해야 함.

- 이 경우 origin과 time zone은 함수에서 요구는 형태와 약간만 달라도, 에러가 발생하는 바, 이에 주의해야 함.

Origin (시간계산 기준일자)	1970년 1월 1일. ※ Default Format 반드시 준수 필요 : "1970-01-01"
Time zone (세계 표준시와 의 차이 반영)	위키피디아의 List of tz database time zones에 나와 있는 TZ명 한국 : "Asia/Seoul" 미국 워싱턴 : "America/Dawson" 일본 : "Asia/Tokyo" 등 ※ GMT : Africa/Abidjan과 동일

Ⅱ. Get & Assign

1. get 함수

특정한 문자열을 지정하여 객체를 호출할 때 사용하는 함수.

※ 호출할 대상이 되는 객체가 메모리상에 올라와 있거나, 그 위치를 지정할 수 있을 경우에만 사용가능

사용 방법

```
get(
    "객체명", # 메모리상에 올라가 있는 R Object를 지칭하는 문자열
    envir=  # "객체명"을 찾을 영역을 별도로 지정할 필요가 있을 경우
)
```

활용 사례

```
for (i in 1:10) {
  set.seed(i)
  assign(paste0("obj_",i), sample(1:10, 7, replace=F))
}
```

2. assign 함수

특정한 문자열을 지정하여 객체를 생성할 때 사용하는 함수.

※ 할당연산자와는 달리, 벡터내의 원소로 객체를 할당하는데는 사용하지 못함.

사용 방법

```
assign(
    "객체명", # 값을 대입하여 생성하고자 하는 객체명(문자열)
     객체      # 작업디렉토리 외부에 있는 데이터를 load할
     envir=  # "객체명"을 찾을 영역을 별도로 지정할 필요가 있을 경우
)
```

활용 사례

```
objdf <- NULL
for (i in 1:10) {
  objdf <- rbind(objdf, get(paste0("obj_",i)))
}
```

Ⅲ. 데이터 조인(Join)

1. match 함수

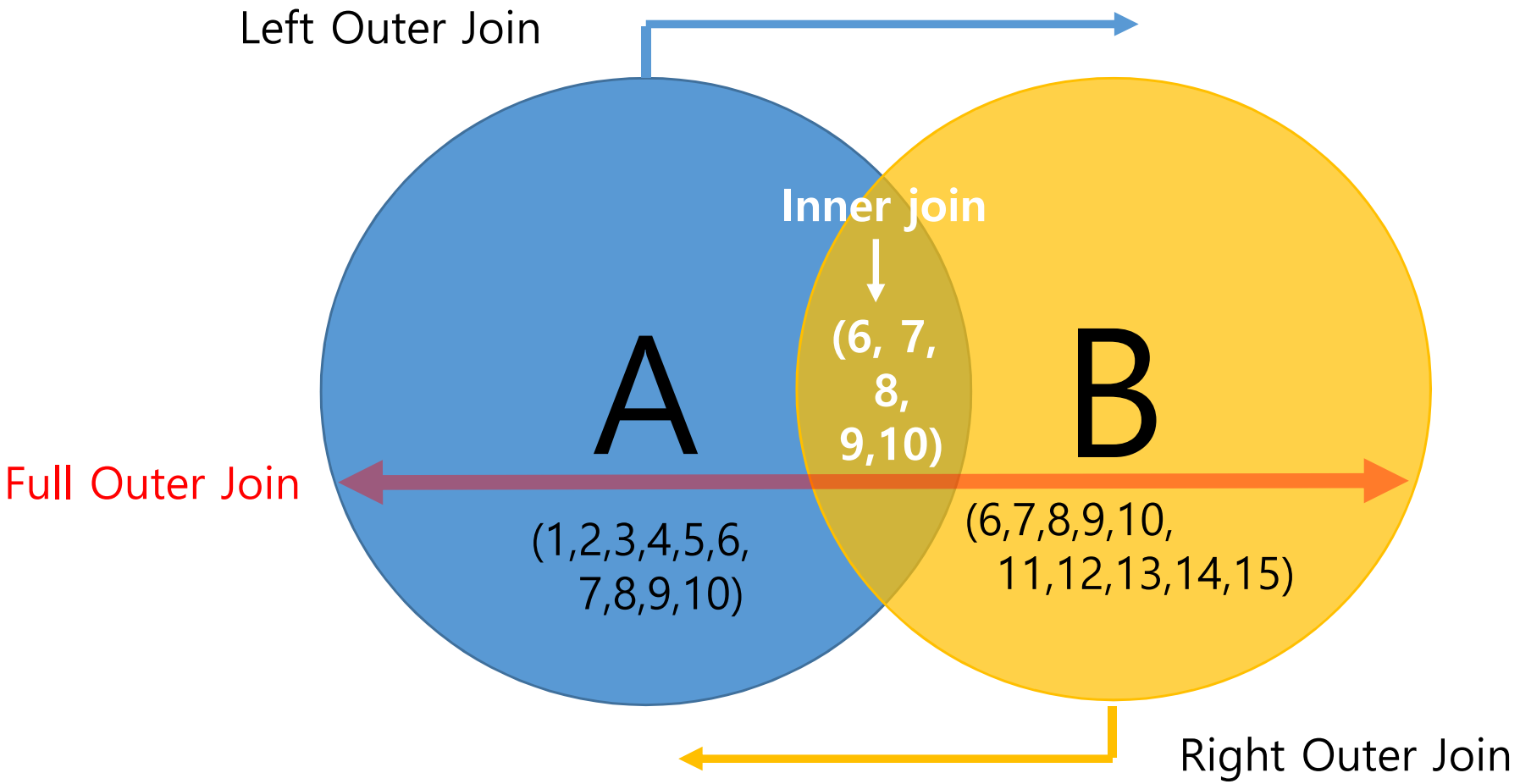
두개의 벡터에서 기준벡터와 일치하는 비교벡터 원소의 자리번호를 리턴해 주는 함수

```
> match(1:10, 7:20) #[1] 0 0 0 0 0 0 1 2 3 4

> match(10:1, 7:20) #[1]  4  3  2  1 NA NA NA NA NA NA

> max(match(1:10, 7:20, nomatch=0))

> intersect <- function(x, y) y[match(x, y, nomatch = 0)]
> intersect(1:10, 7:20)

> intersect_nomatch_1 <- function(x, y) y[match(x, y, nomatch = 1)]
> intersect_nomatch_1(1:10, 7:20)

> intersect_nomatch_NA <- function(x, y) y[match(x, y, nomatch = NA)]
> intersect_nomatch_NA(1:10, 7:20)

> intersect_nomatch_ln <- function(x, y) y[match(x, y, nomatch = length(y))]
> intersect_nomatch_ln(1:10, 7:20)
```

2. 조인(join)의 종류

Join은 크게 내부/외부 조인으로 나뉘고, 외부 조인은 다시 좌/우/Full로 구분됨.

※ 가능한 모든 조합을 찾아내는 Cross-join도 존재함 (A의 원소수 * B의 원소수)



3. 조인(join) 구현 함수

조인을 하기 위해서는 두개의 데이터셋에 동일한 컬럼명을 key로 가지고 있어야 함.

구분		주요 내용	활용함수
내부 조인 Inner Join		두 Table에서 key값 기준 공통부분을 추출	merge, join(... type="inner")
외부 조인 Outer Join	좌측 Left	좌측 Table은 모두 출력하고, 우측 Table은 좌측의 key값과 일치하는 부분만 추출 ※ 좌측 table에 없는 우측 table 관측치는 제외	merge(... all.x=TRUE) join()
	우측 Right	우측 Table은 모두 출력하고, 좌측 Table은 우측의 key값과 일치하는 부분만 추출 ※ 우측 table에 없는 좌측 table 관측치는 제외	merge(... all.y=TRUE) join(... type="right")
	전체 Full	좌측 Table과 우측 Table의 모든 관측치를 추출 ※ 제외되는 관측치가 없음. (단, 중복된 원소는 하나로 합쳐짐)	merge(... all=TRUE) join(... type="full")

자료) (기초를 다지는) 최신 웹개발 공략서 (코가이 단 등 지음, 정인식 옮김)를 요약하여 장표의 일부에 반영.

4. merge 함수

동일한 key를 가진 두개의 데이터프레임에서
key 값이 동일한 관측치만 찾아서 추출해 내는 함수

※ base(기본) 패키지에 포함되어 있는 함수

사용 방법

```
merge( x=필요한 정보를 찾아 붙이고자 하는 대상데이터,
       y= x와 동일한 key열과 새로운 정보를 가지고 있는 데이터,
       all=TRUE, # full outer join
       all.x=TRUE, # left outer join
       all.y=TRUE, # right outer join
       by=NULL,   # cross join
)
```

5. join 함수

Key가 되는 열의 비교를 통하여 동일한 key값을 찾아서,
지정된 열의 값을 옆에다 붙여주는 함수.
※ plyr 패키지에 포함되어 있는 함수임.

사용 방법

```
join( x= # 필요한 정보를 찾아 붙이고자 하는 대상데이터,
      y= # x와 동일한 key열과 새로운 정보를 가지고 있는 데이터,
      by= # key가 되는 열  (x와 y가 동일한 column명을 가져야함.)
      type=c("inner", "left", "right","full")
)
```

End of Document.

감사합니다.