

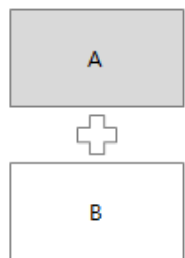

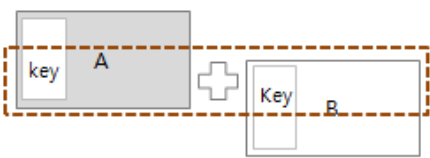
# R 데이터 프레임 결합 : rbind(), cbind(), merge()

분석을 진행하다 보면 하나의 데이터 셋에서 변수를 생성, 제거, 변환하는 작업 못지않게 새로운 데이터 셋을 기존의 데이터 셋과 결합하는 작업 또한 빈번합니다. 이번 포스팅에서는 rbind(), cbind(), merge() 함수를 활용해서 데이터 프레임 결합하는 방법에 대해서 알아보도록 하겠습니다.

예전에 포스팅 했던 R [행렬 함수\(☞ 바로가기\)](#) 에서 rbind(), cbind()를 다루었던 적이 있는데요, 데이터 프레임도 행렬에서의 데이터 결합과 동일하며, 복습하는 차원에서 한번 더 짚어 보고, key값 기준으로 결합하는 merge()에 대해서 추가로 알아보도록 하겠습니다.

## R 데이터 프레임 결합 : rbind(), cbind(), merge()

### [ rbind(), cbind(), merge() 함수 비교 ]

rbind(A, B)	cbind(A, B)	merge(A, B, by='key')
 <b>행 결합</b>	 <b>열 결합</b>	 <b>동일 key 값 기준 결합</b>
		<a href="http://rfriend.tistory.com">http://rfriend.tistory.com</a>

1) 행 결합 (위 + 아래) : rbind(A, B)\*\*

먼저 실습에 사용할 데이터 프레임 두개(cust\_mart\_1, cust\_mart\_2)를 생성해 보겠습니다.

```
## 데이터 프레임 생성
cust_id = c("c01","c02","c03","c04")
last_name = c("Kim", "Lee", "Choi", "Park")
cust_mart_1 = data.frame(cust_id, last_name)
cust_mart_1
cust_mart_2 = data.frame(cust_id = c("c05", "c06", "c07"),
                        last_name = c("Bae", "Kim", "Lim"))
cust_mart_2
```

다음으로 두개의 데이터 프레임(cust\_mart\_1, cust\_mart\_2)을 세로 행 결합 (위 + 아래) 해보도록 하겠습니다.

```
# (1) 행 결합 (위 + 아래) rbind(A, B)
cust_mart_12 = rbind(cust_mart_1, cust_mart_2)
cust_mart_12
```

rbind()는 row bind 의 약자입니다. rbind()를 무작정 외우려고 하지 마시고, row bind의 약자라는걸 이해하시면 됩니다.

**위의 행 결합 rbind()를 하기 위해서는 결합하려는 두개의 데이터 셋의 열의 갯수와 속성, 이름이 같아야만 합니다.**

아래의 예시 처럼 만약 칼럼의 갯수가 서로 다르다면 (cust\_mart\_12는 열이 2개, cust\_mart\_3은 열이 3개) 열의 갯수가 맞지 않는다고 에러 메시지가 뜹니다.

```
cust_mart_3 = data.frame(cust_id = c("c08", "c09"),
                        last_name = c("Lee", "Park"),
                        gender = c("F", "M"))

cust_mart_3
cust_mart_13 = rbind(cust_mart_1, cust_mart_3)
# Error in rbind(deparse.level, ...) :
#   numbers of columns of arguments do not match
```

아래의 예처럼 칼럼의 이름(cust\_mart\_12 는 cust\_id, last\_name 인 반면, cust\_mart\_4는 cust\_id, first\_name)이 서로 다르다면 역시 에러가 납니다.

```
cust_mart_4 = data.frame(cust_id = c("c10", "c11"),
                        first_name = c("Kildong", "Yongpal"))

cust_mart_4
cust_mart_14 = rbind(cust_mart_1, cust_mart_4)
# Error in match.names(cllabs, names(xi)) :
#   names do not match previous names
```

## (2) 열 결합 (왼쪽 + 오른쪽) : cbind(A, B)

```
cust_mart_5 = data.frame(age = c(20, 25, 19, 40, 32, 39, 28),
                        income = c(2500, 2700, 0, 7000, 3400, 3600, 2900))

cust_mart_12
cust_mart_5
cust_mart_125 = cbind(cust_mart_12, cust_mart_5)
cust_mart_125
```

cbind()는 column bind의 약자입니다. **cbind(\*\*)**도 열 결합을 하려고 하면 서로 결합하려는 두 데이터 셋의 관측치가 행이 서로 동일 대상이어야만 하고, 행의 갯수가 서로 같아야만 합니다\*\*.

만약, cbind()를 하는데 있어 행의 갯수가 서로 다르다면 아래의 예처럼 에러 메시지가 뜹니다.

```
cust_mart_6 = data.frame(age = c(34, 50), income = c(3600, 5100))
cust_mart_6
cust_mart_126 = cbind(cust_mart_12, cust_mart_6)
# Error in data.frame(..., check.names = FALSE) :
#   arguments imply differing number of rows: 7, 2
```

### (3) 동일 key 값 기준 결합 : merge(A, B, by='key')

두개의 데이터셋을 열 결합할 때 동일 key 값을 기준으로 결합을 해야 할 때가 있습니다. cbind()의 경우 각 행의 관찰치가 서로 동일 대상일 때 그리고 갯수가 같을 때 가능하다고 했는데요, 만약 각 행의 관찰치가 서로 동일한 것도 있고 그렇지 않은 것도 섞여 있다면 그때는 cbind()를 사용하면 안됩니다. 이때는 동일 key 값을 기준으로 결합을 해주는 merge(A, B, by='key')를 사용해야만 합니다.

아래의 cbind()의 잘못된 예를 하나 보시겠습니다.

```
cust_mart_7 = data.frame(cust_id = c("c03", "c04", "c05", "c06", "c07", "c08", "c09"),
                        buy_cnt = c(3, 1, 0, 7, 3, 4, 1))

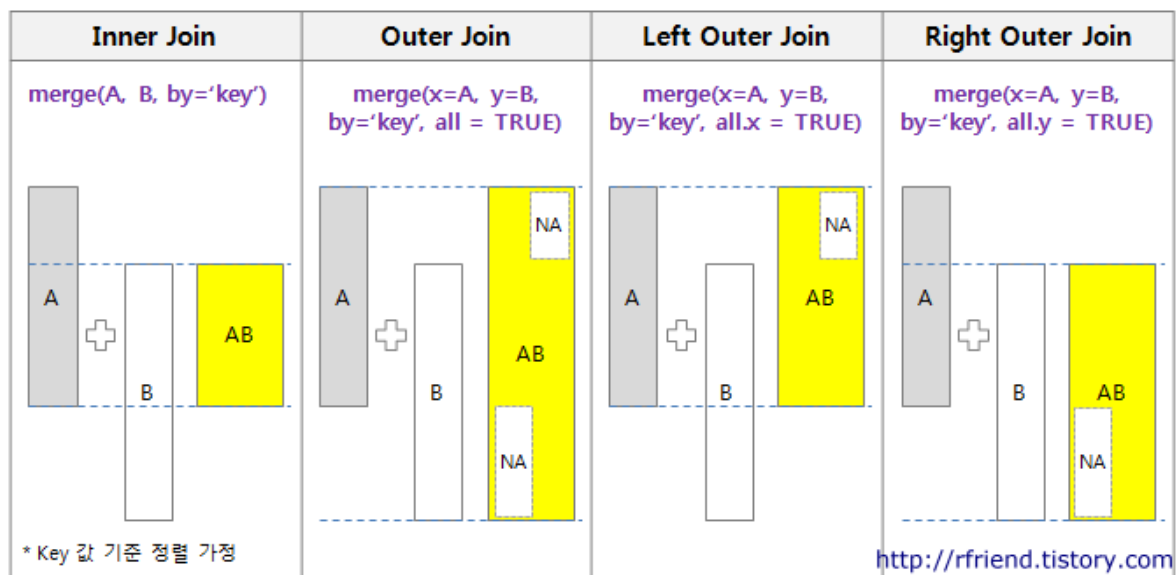
cust_mart_12
cust_mart_7
cust_mart_127 = cbind(cust_mart_12, cust_mart_7)
cust_mart_127
```

cust\_mart\_12와 cust\_mart\_7의 두 개의 데이터 프레임의 관측치가 서로 같은 것(cust\_id가 c03 ~ c07)도 있는 반면, 서로 다른 것(cust\_id가 c01~c02, c08~c09)도 있습니다. 이런 데이터 셋을 cbind()로 결합 시켜버리면 엉뚱한 데이터 셋이 생성되어 버립니다. Oh no~!!!!

이런 경우에는 동일한 key 값을 기준으로 결합을 시켜주는 merge(A, B, by='key')가 답입니다.

SQL에 익숙한 분들은 잘 아시겠지만, merge에는 기준을 어느쪽에 두고 어디까지 포함하느냐에 따라 Inner Join, Outer Join, Left Outer Join, Right Outer Join 등의 4가지 종류가 있습니다. 이를 도식화하면 아래와 같습니다.

#### [ merge() 함수의 join 종류 ]



위에 제시한 4가지 join 유형별로 merge() 함수 사용예를 들어보겠습니다.

#### (3-1) merge() : Inner Join

```
cust_mart_127_innerjoin = merge(x = cust_mart_12, y = cust_mart_7, by =  
'cust_id')  
cust_mart_127_innerjoin
```

### 3-2) merge() - Outer Join\*\*

```
cust_mart_127_outerjoin = merge(x = cust_mart_12, y = cust_mart_7, by =  
'cust_id', all = TRUE)  
cust_mart_127_outerjoin
```

### (3-3) merge() : Left Outer Join

```
cust_mart_127_leftouter <- merge(x = cust_mart_12, y = cust_mart_7, by =  
'cust_id', all.x = TRUE)  
cust_mart_127_leftouter
```

### (3-4) merge() : Right Outer Join

```
cust_mart_127_rightouter <- merge(x = cust_mart_12, y = cust_mart_7, by =  
'cust_id', all.y = TRUE)  
cust_mart_127_rightouter
```

이상 merge() 함수의 4가지 유형의 join 에 대하여 알아보았습니다. 마지막으로, merge() 함수는 2개의 데이터 셋의 결합만 가능하며, **3개 이상의 데이터 셋에 대해서 key 값 기준 merge() 결합을 하려고 하면 에러가** 나는 점 유의하시기 바랍니다.

```
merge(cust_mart_12, cust_mart_5, cust_mart_7, by = 'cust_id')  
# Error in fix.by(by.x, x) :  
#   'by' must specify one or more columns as numbers, names or logical
```

따라서 데이터 프레임 2개씩을 key 값 기준으로 순차적으로 merge() 해나가야 합니다.