

[데이터 처리 성능 향상]

1. plyr 패키지 – xyply 함수들의 예

(1) **adply()** : apply(), sapply() 와 비슷

ex) iris 데이터의 1~4열의 열별 합계 구하기 (apply, adply 사용)

apply(iris[1:4], 2, sum)**adply**(data=iris[,1:4], .margins=2, function(col){ sum(col) }) # margin = 1(행) / 2(열)(2) **ddply()** : summaryBy() 와 비슷

ex) iris 데이터의 꽃받침의 길이와 넓이 각각의 종별 평균 구하기(summaryBy, ddply 사용)

summaryBy(Sepal.Length+Sepal.Width~Species, iris, FUN=c(mean))**ddply**(data=iris, .(Species), function(group){ data.frame(SLmean = mean(group\$Sepal.Length),
SWmean = mean(group\$Sepal.Width)) })

2. 데이터 구조 변경

(1) **melt()** : 특정 열을 창에서 지우기 (지워진 열을 통해 데이터를 출력해도 결과는 실행된다.)(2) **cast**: melt되었던 열 원상복구 - **dcast()**; 변수로 출력 / **acast()**; 행이름으로 출력

3. 데이터 테이블: 짧고 유연한 빠른 구문 사용을 위해 데이터 프레임에서 상속 받음

※ 데이터프레임과 데이터테이블의 함수 사용 비교

	데이터 프레임(data frame)	데이터 테이블(data table)
내림차순 정렬시 - 사용	변수 타입이 숫자인 경우만 사용 가능	변수 타입이 숫자가 아니어도 사용 가능
연달아 이어진 행이나 열 출력	행 번호, 열 번호로만 출력 가능 ex) c[,1:4]	행 및 열번호나 이름으로도 출력 가능 ex) flights_df[,c('year', 'dep_time')]
데이터 빈도 출력	table() 함수 이용	자료명[조건, .(N)] => .N : 개수 출력