

ddply()

ddply()는 데이터 프레임(d)을 입력으로 받아 데이터 프레임(d)을 내보내는 함수다.

▼ 표 5-4 ddply() 함수

① plyr::ddply : 데이터 프레임을 분할하고 ② 함수를 적용한 뒤 ③ 결과를 데이터 프레임으로 반환한다.

```
plyr::ddply(  
  .data,  
  .variables, # 데이터를 그룹 지을 변수명  
  .fun=NULL  
)
```

반환 값은 데이터 프레임이다.

adply()와 ddply()의 가장 큰 차이점이라면 adply()는 행 또는 컬럼 단위로 함수를 적용하는 반면 ddply()는 .variables에 나열한 컬럼에 따라 데이터를 나눈 뒤 함수를 적용한다는 점이다.

다음은 iris 데이터에서 Sepal.Length의 평균을 Species별로 계산하는 예다. 두 번째 인자인 데이터를 그룹 짓는 변수는 .() 안에 기록한다.

```
> ddply(iris,  
+ .(Species),  
+ function(sub) {  
+   data.frame(sepal.width.mean=mean(sub$Sepal.Width))}  
+ )
```

그룹 짓는 변수는 반환되는 데이터 프레임 내 변수로 자동으로 설정된다.

	Species	sepal.width.mean
1	setosa	3.428
2	versicolor	2.770
3	virginica	2.974

새로운 데이터 프레임이 반환된다.

keyname를 설정하지 않았다면, 명령함수 내의 data.frame() 함수를 반드시 사용해야 한다.

여러 변수로 그룹을 짓고자 한다면 .() 안에 조건들 또는 컬럼명들을 콤마로 구분해서 나열한다. 다음은 붓꽃의 종과 Sepal.Length가 5.0보다 큰지 여부 두 가지 조건으로 데이터를 그룹지은 뒤 각 그룹마다 Sepal.Width의 평균을 계산한 예다.

```
> ddply(iris,  
+       .(Species, Sepal.Length > 5.0),  
+       "function(sub)" {  
+         data.frame(sepal.width.mean=mean(sub$Sepal.Width))})
```

	Species	Sepal.Length > 5	sepal.width.mean
1	setosa	FALSE	3.203571
2	setosa	TRUE	3.713636
3	versicolor	FALSE	2.233333
4	versicolor	TRUE	2.804255
5	virginica	FALSE	2.500000
6	virginica	TRUE	2.983673

해당 명령함수의 매개변수에는 data set이 입력된다.

상위 함수 관련

2.3 transform(), summarise(), subset()

← *이런 함수 대신 **비교** 함수를 사용할 수 있음!*

이 절에서 살펴본 예에서는 `adply()` 또는 `ddply()`에 임의의 사용자 정의 함수를 넘겨주어 분석을 수행했다. 그러나 공통적으로 자주 사용되는 유형의 계산은 `transform()`, `summarise()`, `subset()`를 사용해 보다 간단히 표현 할 수 있다.

① transform()

← *즉, 기존 데이터 프레임이 열이 추가된 형태로 변환된다.*

`base::transform()`⁽²⁾은 변수값에 대한 연산결과를 데이터 프레임의 다른 변수에 저장하는 함수이다.

이를 사용해 `baseball` 데이터에 각 행이 선수의 몇년차 통계인지를 뜻하는 `cyear` 컬럼을 추가해보자. 다음 코드는 데이터를 선수 `id` 로 분할 한 뒤, 선수별 분할에서 `year`의 최소값과 현재 행의 `year` 값의 차이를 `cyear`에 저장한다.

```
> head(ddply(baseball, .(id), transform, cyear=year - min(year) + 1))
```

	id	year	stint	team	lg	g	ab	r	h	X2b	X3b	hr
1	aaronha01	1954	1	ML1	NL	122	468	58	131	27	6	13
2	aaronha01	1955	1	ML1	NL	153	602	105	189	37	9	27
3	aaronha01	1956	1	ML1	NL	153	609	106	200	34	14	26
4	aaronha01	1957	1	ML1	NL	151	615	118	198	27	6	44
5	aaronha01	1958	1	ML1	NL	153	601	109	196	34	4	30
6	aaronha01	1959	1	ML1	NL	154	629	116	223	46	7	39

	rbi	sb	cs	bb	so	ibb	hbp	sh	sf	gidp	cyear
1	69	2	2	28	39	NA	3	6	4	13	1
2	106	3	1	49	61	5	3	7	4	20	2
3	92	2	4	37	54	6	2	5	7	21	3
4	132	1	1	57	58	15	0	0	3	13	4
5	95	4	1	59	49	16	1	0	3	21	5
6	123	8	0	51	54	17	4	0	9	19	6

②

summarise()

`plyr::summarise()`는 데이터의 요약 정보를 만드는데 사용하는 함수이다. `transform()`이 인자로 주어진 계산 결과를 새로운 컬럼에 추가한 데이터 프레임을 반환하는 반면 `summarise()`는 계산 결과를 담은 새로운 데이터 프레임을 반환한다.

baseball 데이터에서 각 선수의 최초 데이터가 몇년도에 해당하는지 살펴보는 다음 예를 보자. 아래 코드에서는 각 id 마다 최소 year를 minyear로 갖는 데이터 프레임들이 `summarise()`에 의해 생성되고 `ddply`는 이들 데이터 프레임을 모아 하나의 데이터 프레임으로 반환한다.

```
> head(ddply(baseball, .(id), summarise, minyear=min(year)))
      id minyear
1 aaronha01  1954
2 abernte02  1955
3 adairje01  1958
4 adamsba01  1906
5 adamsbo03  1946
6 adcocjo01  1950
```

만약 여러 계산값들을 구하고 싶다면 인자를 계속 나열하면 된다. 다음은 minyear, maxyear를 구하는 예이다.

```
> head(ddply(baseball, .(id), summarise,
+           minyear=min(year), maxyear=max(year)))
      id minyear maxyear
1 aaronha01  1954   1976
2 abernte02  1955   1972
```

128

plyr 패키지

```
3 adairje01  1958   1970
4 adamsba01  1906   1926
5 adamsbo03  1946   1959
6 adcocjo01  1950   1966
```