**1. data.table?**

R에는 data handling을 위한 여러가지 패키지들이 있다. 그중 data.table패키지는 가장 많이 사용하는 패키지라 할 수 있다. data.table은 큰 데이터를 탐색, 연산, 병합 하는데 아주 유용하다.

data.table의 빠른 속도의 원리는 특정 column을 key값으로 **색인**을 지정한 후 데이터를 처리한다.

따라서 기존의 방식(data.frame)보다 월등히 빠른 속도를 보여준다.

현재 data.table 패키지 버전은 1.8.10까지 나와 있으며 R버전 (≥ 2.12.0) 이상부터 지원한다.

**2. data.table vs data.frame**

여기서 사용하는 데이터 샘플은 titanic data로 [여기](https://drive.google.com/file/d/0ByvctC8PhhQncFQ5UVpFVUp1ZEk/edit?usp=sharing)서 다운로드 받을 수 있다.

**가. Data load**

library(data.table)

titanic <- read.csv("titanic.csv")

# class 변환

titanic.dt <- data.table(titanic)

class(titanic)

## [1] "data.frame"

class(titanic.dt)

## [1] "data.table" "data.frame"

**나. 속도의 차이**

data.table을 사용하는 가장 큰 이유는 기존 data.frame에서 하는 작업보다 최소 **20**배가량 빠르다.

data.frame과의 속도를 비교해보았다.

DF <- data.frame(x = runif(2.6e+07), y = rep(LETTERS, each = 10000))

df <- data.frame(x = runif(2.6e+07), y = rep(letters, each = 10000))

system.time(x <- DF[DF$y == "C", ])

## user system elapsed

## 3.59 0.18 3.76

DT <- as.data.table(DF)

setkey(DT, y)

system.time(x <- DT[J("C"), ])

## user system elapsed

## 0.04 0.00 0.03

결과를 보면 100배 이상 빠른걸 볼 수 있다. 데이터가 많을 수록 속도는 더 차이가 나타난다.

**다. 데이터 선택**

data.table의 형식은 [행, 표현식, 옵션]으로 되어 있으며 data.frame과는 조금 다르다.

그 차이점을 아래에서 볼 수 있다.

titanic.dt[1]

## X pclass survived name sex age sibsp parch

## 1: 1 1st 1 Allen, Miss. Elisabeth Walton female 29 0 0

## ticket fare cabin embarked boat body home.dest

## 1: 24160 211.3 B5 Southampton 2 NA St Louis, MO

1열를 data.frame에서 가져온다면 [,1]을 사용하지만 data.table에서는 다르다.

head(titanic[, 1]) #(data.frame)

## [1] 1 2 3 4 5 6

data.table에서는 다른 결과를 보인다.

titanic.dt[, 1] #(data.table)

## [1] 1

옵션 **with=F** 를 써주면 data.frame에서 사용하던 방식을 사용할 수 있다.

titanic.dt[, 1, with = F]

## X

## 1: 1

## 2: 2

## 3: 3

## 4: 4

## 5: 5

## ---

## 1305: 1305

## 1306: 1306

## 1307: 1307

## 1308: 1308

## 1309: 1309

열 번호대신 변수명을 쓰면 data.frame과 같은 결과를 얻을 수 있다.

head(titanic.dt[, X]) #열 변호대신 변수명 사용

## [1] 1 2 3 4 5 6

**3. use a data.table**

**가. 조건으로(제약) 데이터 선택**

data.table에서느 **J 표현식**이 있다. **dt[J('제약조건')]**으로 작성할 수 있으며 **J** 를 사용할 경우 **setkey**가 꼭 지정되어

있어야한다. 사용방법을 확인해보자.

데이터에서 1등급 승객 리스트를 뽑아보면 아래와 같으며 두 코드 모두 동일한 결과를 보인다.

titanic.dt[pclass == "1st"]

titanic.dt[pclass == "1st", ]

## X pclass survived name sex age

## 1: 1 1st 1 Allen, Miss. Elisabeth Walton female 29.0000

## 2: 2 1st 1 Allison, Master. Hudson Trevor male 0.9167

## 3: 3 1st 0 Allison, Miss. Helen Loraine female 2.0000

## 4: 4 1st 0 Allison, Mr. Hudson Joshua Crei male 30.0000

## 5: 5 1st 0 Allison, Mrs. Hudson J C (Bessi female 25.0000

## ---

## 319: 319 1st 0 Williams-Lambert, Mr. Fletcher male NA

## 320: 320 1st 1 Wilson, Miss. Helen Alice female 31.0000

## 321: 321 1st 1 Woolner, Mr. Hugh male NA

## 322: 322 1st 0 Wright, Mr. George male 62.0000

## 323: 323 1st 1 Young, Miss. Marie Grice female 36.0000

## ... 생략 ...

**key**를 이용하여 데이터를 가져올 수 도 있다.

setkey(titanic.dt, pclass)

# data.table 정보 확인

tables()

## NAME NROW MB

## [1,] DT 26,000,000 298

## [2,] titanic.dt 1,309 1

## [3,] x 1,000,000 12

## COLS

## [1,] x,y

## [2,] X,pclass,survived,name,sex,age,sibsp,parch,ticket,fare,cabin,embarked,boat,body,

## [3,] y,x

## KEY

## [1,] y

## [2,] pclass

## [3,] y

## Total: 311MB

titanic.dt[J("1st")]

## pclass X survived name sex age

## 1: 1st 1 1 Allen, Miss. Elisabeth Walton female 29.0000

## 2: 1st 2 1 Allison, Master. Hudson Trevor male 0.9167

## 3: 1st 3 0 Allison, Miss. Helen Loraine female 2.0000

## 4: 1st 4 0 Allison, Mr. Hudson Joshua Crei male 30.0000

## 5: 1st 5 0 Allison, Mrs. Hudson J C (Bessi female 25.0000

## ---

## 319: 1st 319 0 Williams-Lambert, Mr. Fletcher male NA

## 320: 1st 320 1 Wilson, Miss. Helen Alice female 31.0000

## 321: 1st 321 1 Woolner, Mr. Hugh male NA

## 322: 1st 322 0 Wright, Mr. George male 62.0000

## 323: 1st 323 1 Young, Miss. Marie Grice female 36.0000

## ... 생략 ...

계산도 동시에 할 수 있다. :)

titanic.dt[pclass == "1st", mean(survived)]

## [1] 0.6192

titanic.dt[J("1st"), mean(survived)]

## pclass V1

## 1: 1st 0.6192

**나. data group 연산**

**dt[,조건, by='variable']** 표현식으로 사용가능하다.

등급별 생존률

titanic.dt[, mean(survived), by = "pclass"]

## pclass V1

## 1: 1st 0.6192

## 2: 2nd 0.4296

## 3: 3rd 0.2553

1등급 승객의 성별 생존률

titanic.dt[J("1st"), mean(survived), by = "sex"]

## sex V1

## 1: female 0.9653

## 2: male 0.3408

group key는 여러개 지정가능하다.

titanic.dt[J("1st"), mean(survived), by = "sex,boat"]

## sex boat V1

## 1: female 2 1.0000

## 2: male 11 1.0000

## 3: female 0.1667

## 4: male 0.0000

## 5: male 3 1.0000

## 6: female 10 1.0000

## ... 생략 ...

## sex boat V1

1등급 승객의 성비율을 알고 있다면

titanic.dt[pclass == "1st", length(which(age > 20))/.N, by = "sex"]

## sex V1

## 1: female 0.8125

## 2: male 0.7877

counting(1등급 승객의 카운팅)

titanic.dt[, length(which(pclass == "1st"))]

## [1] 323

titanic.dt[pclass == "1st", .N]

## [1] 323

group counting(1등급 승객의 성별 카운팅)

titanic.dt[pclass == "1st", .N, by = "sex"]

## sex N

## 1: female 144

## 2: male 179

titanic.dt[, .N, by = "pclass"]

## pclass N

## 1: 1st 323

## 2: 2nd 277

## 3: 3rd 709

1등급 승객 성인의 비율을 성별로 나누어 계산

titanic.dt[pclass == "1st", length(which(age > 20))/nrow(.SD), by = "sex"]

## sex V1

## 1: female 0.8125

## 2: male 0.7877

titanic.dt[pclass == "1st", length(which(age > 20))/.N, by = "sex"]

## sex V1

## 1: female 0.8125

## 2: male 0.7877

**다. data merge 연산**

data.frame의 merge보다 약 400배 이상 빠른 속도를 보인다.

data.frame merge

# data.frame 생성

DF <- data.frame(x = runif(2.6e+07), y = rep(LETTERS, each = 10000))

df <- data.frame(y = c("A", "B", "C"), z = c("a", "b", "c"))

# data.table 생성

DT <- as.data.table(DF)

dt <- as.data.table(df)

# 데이터확인

head(DF)

## x y

## 1 0.5853 A

## 2 0.6798 A

## 3 0.5041 A

## 4 0.9614 A

## 5 0.4453 A

## 6 0.8605 A

head(df)

## y z

## 1 A a

## 2 B b

## 3 C c

# data.frame 시간측정

system.time(tmp.merge <- merge(DF, df, by = "y"))

## user system elapsed

## 29.13 0.95 30.09

data.table merge

# 먼저 setkey를 이용하여 key값을 지정

setkey(DT, y)

system.time(tmp.merge.dt <- DT[dt, ])

## user system elapsed

## 0.07 0.00 0.06

월등한 속도차이를 보인다.

**라. data 수정 및 삭제**

**DT[i, 새로운 변수명**`**=**`**값]** 표현식으로 data에서 계산을하거나 사용자가 원하는 새로운 변수를 만들수 있다.

titanic data에 isminor란 변수에 '**adult**'라는 값을 추가

titanic.dt[, `:=`(isminor, "adult")]

## X pclass survived name sex age sibsp

## 1: 1 1st 1 Allen, Miss. Elisabeth Walton female 29.0000 0

## 2: 2 1st 1 Allison, Master. Hudson Trevor male 0.9167 1

## 3: 3 1st 0 Allison, Miss. Helen Loraine female 2.0000 1

## 4: 4 1st 0 Allison, Mr. Hudson Joshua Crei male 30.0000 1

## 5: 5 1st 0 Allison, Mrs. Hudson J C (Bessi female 25.0000 1

## 6: 6 1st 1 Anderson, Mr. Harry male 48.0000 0

## parch ticket fare cabin embarked boat body

## 1: 0 24160 211.34 B5 Southampton 2 NA

## 2: 2 113781 151.55 C22 C26 Southampton 11 NA

## 3: 2 113781 151.55 C22 C26 Southampton NA

## 4: 2 113781 151.55 C22 C26 Southampton 135

## 5: 2 113781 151.55 C22 C26 Southampton NA

## 6: 0 19952 26.55 E12 Southampton 3 NA

## home.dest isminor

## 1: St Louis, MO adult

## 2: Montreal, PQ / Chesterville, ON adult

## 3: Montreal, PQ / Chesterville, ON adult

## 4: Montreal, PQ / Chesterville, ON adult

## 5: Montreal, PQ / Chesterville, ON adult

## 6: New York, NY adult

추가된 data 확인

titanic.dt[isminor == "adult", .N]

## [1] 1309

age가 15이하일때는 '**child**'로 값은 수정

titanic.dt[age < 15, `:=`(isminor, "child")]

## X pclass survived name sex age

## 1: 1 1st 1 Allen, Miss. Elisabeth Walton female 29.0000

## 2: 2 1st 1 Allison, Master. Hudson Trevor male 0.9167

## 3: 3 1st 0 Allison, Miss. Helen Loraine female 2.0000

## 4: 4 1st 0 Allison, Mr. Hudson Joshua Crei male 30.0000

## 5: 5 1st 0 Allison, Mrs. Hudson J C (Bessi female 25.0000

## ---

## 1305: 1305 3rd 0 Zabour, Miss. Hileni female 14.5000

## 1306: 1306 3rd 0 Zabour, Miss. Thamine female NA

## 1307: 1307 3rd 0 Zakarian, Mr. Mapriededer male 26.5000

## 1308: 1308 3rd 0 Zakarian, Mr. Ortin male 27.0000

## 1309: 1309 3rd 0 Zimmerman, Mr. Leo male 29.0000

## sibsp parch ticket fare cabin embarked boat body

## 1: 0 0 24160 211.337 B5 Southampton 2 NA

## 2: 1 2 113781 151.550 C22 C26 Southampton 11 NA

## 3: 1 2 113781 151.550 C22 C26 Southampton NA

## 4: 1 2 113781 151.550 C22 C26 Southampton 135

## 5: 1 2 113781 151.550 C22 C26 Southampton NA

## ---

## 1305: 1 0 2665 14.454 Cherbourg 328

## 1306: 1 0 2665 14.454 Cherbourg NA

## 1307: 0 0 2656 7.225 Cherbourg 304

## 1308: 0 0 2670 7.225 Cherbourg NA

## 1309: 0 0 315082 7.875 Southampton NA

## home.dest isminor

## 1: St Louis, MO adult

## 2: Montreal, PQ / Chesterville, ON child

## 3: Montreal, PQ / Chesterville, ON child

## 4: Montreal, PQ / Chesterville, ON adult

## 5: Montreal, PQ / Chesterville, ON adult

## ---

## 1305: child

## 1306: adult

## 1307: adult

## 1308: adult

## 1309: adult

수정된 data 확인

titanic.dt[isminor == "adult", .N]

## [1] 1200

titanic.dt[isminor == "child", .N]

## [1] 109

계산결과 값으로도 가능, 요금을 원화로 계산

titanic.dt[, `:=`(fare.won, fare \* 1000)]

## X pclass survived name sex age sibsp

## 1: 1 1st 1 Allen, Miss. Elisabeth Walton female 29.0000 0

## 2: 2 1st 1 Allison, Master. Hudson Trevor male 0.9167 1

## 3: 3 1st 0 Allison, Miss. Helen Loraine female 2.0000 1

## 4: 4 1st 0 Allison, Mr. Hudson Joshua Crei male 30.0000 1

## 5: 5 1st 0 Allison, Mrs. Hudson J C (Bessi female 25.0000 1

## 6: 6 1st 1 Anderson, Mr. Harry male 48.0000 0

## parch ticket fare cabin embarked boat body

## 1: 0 24160 211.34 B5 Southampton 2 NA

## 2: 2 113781 151.55 C22 C26 Southampton 11 NA

## 3: 2 113781 151.55 C22 C26 Southampton NA

## 4: 2 113781 151.55 C22 C26 Southampton 135

## 5: 2 113781 151.55 C22 C26 Southampton NA

## 6: 0 19952 26.55 E12 Southampton 3 NA

## home.dest isminor fare.won

## 1: St Louis, MO adult 211337

## 2: Montreal, PQ / Chesterville, ON child 151550

## 3: Montreal, PQ / Chesterville, ON child 151550

## 4: Montreal, PQ / Chesterville, ON adult 151550

## 5: Montreal, PQ / Chesterville, ON adult 151550

## 6: New York, NY adult 26550