

[R] R에서 사용되는 Data.frame 과 Factor 에 사용되는 다양한 함수

노트북: [TIL-MY]

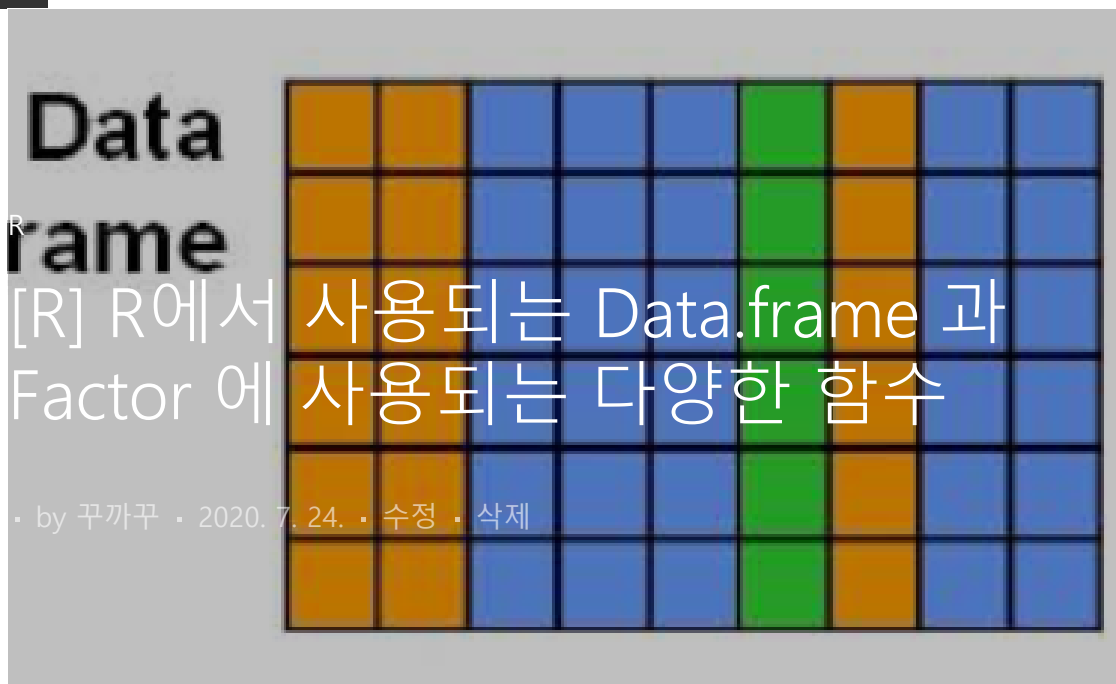
만든 날짜: 2020-07-24 오전 8:15

URL: <https://continuous-development.tistory.com/36>

나무늘보의 개발 블로그

홈

태그



#데이터 프레임(data.frame)

분류 전체보기 

Python

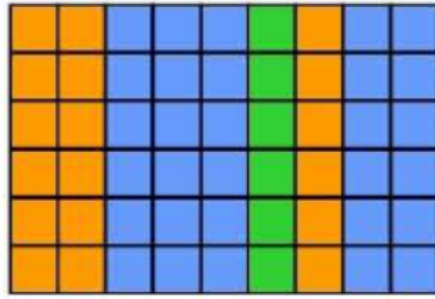
Database 

ASP.NET

Algorithm

Deep learning

Data Frame



배열(array) 3차원 벡터로서 동일 타입의 데이터만 저장 가능하다.

2차원 구조로서 열 단위로 서로 다른 타입의 데이터들로 구성 가능하다.

또한 모든 열의 데이터 개수(행의 개수)는 동일해야 한다.

데이터 프레임 변환 :rbind(df, 벡터), cbind(df, 벡터)

데이터 프레임의 구조 확인 :str(df)

인덱싱 : [행의인덱싱, 열의 인덱싱], [열의 인덱싱], df\$컬럼 이름, [[열 인덱싱]]

subset (df, select=컬럼명들, subset=(조건))

AWS

ETC..

R 

공지사항

글 보실 때 주의사항

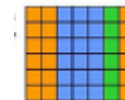
: 최근글 : 인기글

[R] R로 ...



2020.07.24

[R] R에 ...



2020.07.24

[R] R에 ...



2020.07.23

[R] R에 ...



2020.07.23

[R] R에 ...



2020.07.22

최근댓글

태그

테이블 생성,
날짜함수,

data.frame(value1,value2) - 데이터 프레임 생성

```
407 ~ #####
408 #data.frame
409 #행렬과 비슷하다
410 #다만 , 다양한 변수 (관측값이 숫자, 문자 , 범주 등) 으로 표현된다.
411 #각 열에 대한 접근은 $이용하여 접근할 수 있다.
412 # 인덱스를 활용하는 방법도 있다.
413
414 x <- c(1,3,5,7,9)
415 y <- c(2,4,6,8,10)
416
417 exampleDF <- data.frame(x,y)
418 exampleDF
419 str(exampleDF)
420
421 exampleDF[1,]
422 exampleDF[,c("x")]
423 class(exampleDF[,c("x")])
424 exampleDF[
425
```

```
3 5 6
4 7 8
5 9 10
> str(exampleDF)
'data.frame': 5 obs. of 2 variables:
 $ x: num 1 3 5 7 9
 $ y: num 2 4 6 8 10
>
> exampleDF[1,]
  x y
1 1 2
> exampleDF[,c("x")]
[1] 1 3 5 7 9
> class(exampleDF[,c("x")])
[1] "numeric"
> exampleDF
  x y
1 1 2
2 3 4
3 5 6
4 7 8
5 9 10
```

#colnames()/rownames() - 행 / 열 이름 변경

R 정규표현식 사용법,
SQL, DDL,
array 함수,
인스턴스, strsplit,
R 정규표현식,
R 배열, rbind,
str_extract_all,
AWS,
Oracle SQL,
cbind함수,
do.call, R 제어문,
Oracle, R FOR,
행렬, colnames,
substr,
rownames,
matrix함수, 설정,
cbind, R IF,
unlist함수,
사용법,
rbind함수

전체 방문자

105

Today : 1

Yesterday : 2

```
426 # colnames(), rownames()
427 colnames(exampleDF) <- c("val01","val02")
428 exampleDF
429
430 colist <- names(exampleDF)
431 class(colist)

414:18 # (Untitled) ⌵

Console Terminal × Jobs ×

~/ ➜
> # colnames(), rownames()
> colnames(exampleDF) <- c("val01","val02")
> exampleDF
  val01 val02
1     1     2
2     3     4
3     5     6
4     7     8
5     9    10
>
> colist <- names(exampleDF)
> class(colist)
[1] "character"
> |
```

예제

```
433 # 문자열 벡터 , 숫자형 벡터 , 문자열 벡터
434 # data.frame
435
436 stuName <- c("조동균","한소연","박수진","최가은")
437 subject.eng <- c(100,100,100,70)
438 subject.math <- c(80,75,100,100)
439 subject.kor <- c(100,100,100,70)
440 score.grade <- c("A","B","A","c")
441
442 student <- data.frame(stuName,subject.eng,subject.math,subject.kor,score.grade)
443 student
444 colnames(student) <- c("이름","영어","수학","국어","성적")
445 student
446

422:21 # (Untitled) ⌵

Console Terminal × Jobs ×

~/ ➜
> stuName <- c("조동균","한소연","박수진","최가은")
> subject.eng <- c(100,100,100,70)
> subject.math <- c(80,75,100,100)
> subject.kor <- c(100,100,100,70)
> score.grade <- c("A","B","A","c")
>
> student <- data.frame(stuName,subject.eng,subject.math,subject.kor,score.grade)
> student
  stuName subject.eng subject.math subject.kor score.grade
1 조동균         100          80         100          A
2 한소연         100          75         100          B
3 박수진         100         100         100          A
4 최가은          70         100          70          c
> colnames(student) <- c("이름","영어","수학","국어","성적")
> student
   이름  영어  수학  국어  성적
1 조동균  100   80  100   A
2 한소연  100   75  100   B
3 박수진  100  100  100   A
4 최가은   70  100   70   c
>
```

#nrow(value) - 행의 갯수 출력

```
447 |
448 # nrow() - 행의 갯수
449 nrow(student)
450 ncol(student)
451
447:1 # (Untitled)
Console Terminal x Jobs x
~/
> # nrow() - 행의 갯수
> nrow(student)
[1] 4
> ncol(student)
[1] 5
```

```
452 |
453 # 열 추가
454 # 학생의 학번의 열을
455 학번<-c("056541","995254","205477","182354")
456
457 scondschDF <- cbind(student ,학번)
458 scondschDF
459 |
460
459:1 # (Untitled)
Console Terminal x Jobs x
~/
> # 열 추가
> # 학생의 학번의 열을
> 학번<-c("056541","995254","205477","182354")
> scondschDF <- cbind(student ,학번)
> scondschDF
  이름 영어 수학 국어 성적 학번
1 조동균 100  80  100  A 056541
2 한소연 100  75  100  B 995254
3 박수진 100 100  100  A 205477
4 최가은  70 100  70   c 182354
```

#cbind(value1, value2) - 열 추가 함수

#rbind(value1, value2) - 행 추가 함수

```
447
448 # nrow() - 행의 갯수
449 nrow(student)
450 ncol(student)
451
452
453 # 열 추가
454 # 학생의 학번의 열을 추가
455 학번<-c("056541","995254","205477","182354")
456
457 scondschDF <- cbind(student ,학번)
458 scondschDF
459 str(scondschDF)
460
461 #터미 데이터를 이용해서 행 추가
462 newStudent<-c("라이언","100","100","100","A+","777777")
463
464 thrdschDF <- rbind(scondschDF, newStudent)
465 thrdschDF
466 thrdschDF$이름
467 thrdschDF$이름[5]
468 thrdschDF[[1]][5]
469 thrdschDF[1:4]
470
471
```

449:14 [Untitled] :

Console Terminal Jobs

```
~/
3 박수진 100 100 100 A 205477
4 최가은 70 100 70 c 182354
> str(scondschDF)
'data.frame': 4 obs. of 6 variables:
 $ 이름: chr "조동균" "관소연" "박수진" "최가은"
 $ 영어: num 100 100 100 70
 $ 수학: num 80 75 100 100
 $ 국어: num 100 100 100 70
 $ 성적: chr "A" "B" "A" "C"
 $ 학번: chr "056541" "995254" "205477" "182354"
>
> #터미 데이터를 이용해서 행 추가
> newStudent<-c("라이언","100","100","100","A+","777777")
>
> thrdschDF <- rbind(scondschDF, newStudent)
> thrdschDF
  이름 영어 수학 국어 성적 학번
1 조동균 100 80 100 A 056541
2 관소연 100 75 100 B 995254
3 박수진 100 100 100 A 205477
4 최가은 70 100 70 c 182354
5 라이언 100 100 100 A+ 777777
>
```

#with(data, expression) - 데이터 프레임 또는 리스트 내 필드를 필드 이름만으로 접근할 수 있게 해주는 함수

#within(data, expression) - with함수의 기능에 더해서 데이터를 수정하는 기능까지 제공

```
483 # with(data, expression) ~ 여러가지 값을 한번에 확인 ,within(data, expression) ~ 값을 확인해서 다시 반영하는 용도
484
485 data(iris)
486 iris
487
488 mean(iris$Sepal.Length)
489 mean(iris$Sepal.Width )
490
491 with( # $를 쓰지 않고 바로 사용한다 왜냐하면 within에 iris가 들어가 있어서 이미 접근한 상태이다.
492   iris,
493   {
494     print(mean(Sepal.Length))
495     print(mean(Sepal.Width))
496   }
497 )
498
499
500 x <- data.frame(val=c(1,2,3,4,NA,5,NA))
501 x
502
503 x<-within(
504   x,
505   {
506     val <- ifelse(is.na(val),mean(val,na.rm=T),val )
507   }
508 )
509 x
510
```

49529 (Untitled) :

Console Terminal Jobs

```
> mean(iris$Sepal.Length)
[1] 5.843333
> mean(iris$Sepal.Width )
[1] 3.057333
> with( # $를 쓰지 않고 바로 사용한다 왜냐하면 within에 iris가 들어가 있어서 이미 접근한 상태이다.
+   iris,
+   {
+     print(mean(Sepal.Length))
+     print(mean(Sepal.Width))
+   }
+ )
[1] 5.843333
[1] 3.057333
> x <- data.frame(val=c(1,2,3,4,NA,5,NA))
> x
  val
1   1
2   2
3   3
4   4
5  NA
6   5
7  NA
```

```
> x<-within(
+   x,
+   {
+     val <- ifelse(is.na(val),mean(val,na.rm=T),val )
+   }
+ )
> x
  val
1   1
2   2
3   3
4   4
5   3
6   5
7   3
```

#within을 통해 결측치값을 평균으로 바꿔주는 구문

```

34
35 irisSLMedian<- sapply(split(iris$Sepal.Length,iris$Species)
36                       ,median
37                       ,na.rm=T )
38
39 class(irisSLMedian)
40 irisSLMedian[iris$Species]
41
42 iris<-within(
43   iris,
44   {
45     Sepal.Length <- ifelse(is.na(Sepal.Length),irisSLMedian[iris$Species],Sepal.Length )
46   }
47 )
48 iris
49
50 iris2<-within(
51   iris,
52   {
53     Sepal.Length <- ifelse(is.na(Sepal.Length), sapply(split(iris$Sepal.Length,iris$Species)
54                                                         ,median
55                                                         ,na.rm=T ) ,Sepal.Length )
56   }
57 )
58 iris2
59
60
61

```

#split(feature,분류기준,[중위값],[결측값을 중앙값으로 변환]) - 분류기준에 따라 데이터를 나누어 반환하는 함수

```

31
32 #split( feature , 분류기준 ) - 분류기준에 따라 나누는 함수
33 #na.rm은 결측값을 제거하고 중위수를 구하겠다는 말이다.
34
35 sapply(split(iris$Sepal.Length,iris$Species)
36        ,median
37        ,na.rm=T )
38
39
40

```

41:1 (Top Level) ↕

Console Terminal × Jobs ×

~/ ↗

```

> sapply(split(iris$Sepal.Length,iris$Species)
+       ,median
+       ,na.rm=T )
  setosa versicolor  virginica
    5.0         5.9         6.5
> |

```


#subset(value, 조건, [select]) - 설정하는 조건에 맞는 벡터, 매트릭스 혹은 데이터 프레임을 반환하는 함수

```
63 #subset() - 설정하는 조건에 맞는 벡터, 매트릭스 혹은 데이터 프레임을 반환
64 ?subset
65
66 x <- 1:5
67 y <- 6:10
68
69 ?letters
70
71 z <- letters[1:5]
72 z
73
74 exampleDF <- data.frame(x,y,z)
75 str(exampleDF)
76
77 # x의 값이 3이상인 결과를 새로운 데이터 프레임으로 만들어보자
78 subDF01<- subset(exampleDF , x >= 3)
79 subDF01
80
81 # y의 값이 8이하인 결과를 새로운 데이터 프레임으로 만들어 보자
82 subDF02<- subset(exampleDF , y <= 8)
83 subDF02
84
85 # x의 값이 2이상이고 y의 값이 8이하인 결과를 새로운 데이터 프레임으로 만들어 보자
86 subDF03<- subset(exampleDF, x >= 2 & y <= 8)
87 subDF03
88
89 #subset()에서 select 으로 원하는 컬럼 선택 가능
90 subDF2 <- subset(exampleDF, x>=3, select=c(x,y))
91 subDF2
```

#select 조건 - 원하는 컬럼만 가져온다.

```
91 #subset()에서 select 으로 원하는 컬럼 선택 가능
92 subDF2 <- subset(exampleDF, x>=3, select=c(x,y))
93 subDF2
94
95 iris
96 str(iris)
97 names(iris)
98
99 # Petal.Length 평균을 구하라
100 mean(iris$Petal.Length)
101 iris_df <- subset(iris, Petal.Length >= mean(Petal.Length) , select=c(Sepal.Length,Petal.Length,Species ))
102 iris_df
103
104 str(iris_df)
105 |
106
1051 (Top Level) :

```

Console	Terminal	Jobs
~/		
440	6.9	5.4 virginica
441	6.7	5.6 virginica
442	6.9	5.1 virginica
443	5.8	5.1 virginica
444	6.8	5.9 virginica
445	6.7	5.7 virginica
446	6.7	5.2 virginica
447	6.3	5.0 virginica
448	6.5	5.2 virginica
449	6.2	5.4 virginica
450	5.9	5.1 virginica

```
str(iris_df)
data.frame': 93 obs. of 3 variables:
 $ Sepal.Length: num 7 6.4 6.9 5.5 6.5 5.7 6.3 6.6 5.2 5.9 ...
 $ Petal.Length: num 4.7 4.5 4.9 4 4.6 4.5 4.7 4.6 3.9 4.2 ...
 $ Species : Factor w/ 3 levels "setosa","versicolor",...: 2 2 2 2 2 2 2 2 2 2 ...
```

#Factor - 범주형 변수를 나타낸다.

as.factor / factor 로 생성할 수 있다.

```
106 ## Factor?
107 # 범주형 변수
108 ?factor
109 gender <- factor("m",c("m","f"))
110 gender
111
112 #레벨의 갯수를 확인할 수 있다.
113 nlevels(gender)
114 #구성되어 있는 레벨을 출력해준다.
115 levels(gender)
116 levels(gender)[1]
117
118 blood.type <- factor(c("A","A","AB","D","D"))
119 blood.type[6] <- "D" #범주에 없는 값을 넣지 못한다.
120
121 is.factor(blood.type) #factor형인지 확인하는 함수
122
123 lettersVec <- c("a","b","b","c","a","c","a","a","a")
124 lettersVec
125 class(lettersVec)
126 lettersVec.fac <- as.factor(lettersVec) # factor 형으로 변환
127 lettersVec.fac
128
129 lettersVec.fac <- factor(lettersVec, # factor를 생성하는데 이때는 라벨이랑 레벨이 가능하다
130                          levels = c("a","b","c"),
131                          labels = c("best","middle","low"))
132
133 lettersVec.fac
134
135
101:94 (Top Level) :
Console Terminal Jobs
~/
> lettersVec.fac <- as.factor(lettersVec) # factor 형으로 변환
> lettersVec.fac
[1] a b b c a c a a
Levels: a b c
> lettersVec.fac <- factor(lettersVec, # factor를 생성하는데 이때는 라벨이랑 레벨이 가능하다
+                          levels = c("a","b","c"),
+                          labels = c("best","middle","low"))
> lettersVec.fac
[1] best middle middle low best low best best best
Levels: best middle low
> |
```

factor로 형변환

```
136
137 id     <- c(1,2,3,4,5)
138 gender  <- c("F","M","F","M","F")
139
140 data <- data.frame(id = id,gender = gender)
141 data
142 str(data)
143
144 data$gender <- as.factor(gender)
145 str(data)
146 levels(data$gender) <- c("female","male")
147 str(data)
148 data
149
148:1 (Top Level) ↕
```

Console Terminal × Jobs ×

~/ ➡

```
4 4 M
5 5 F
> str(data)
'data.frame': 5 obs. of 2 variables:
 $ idx : num 1 2 3 4 5
 $ gender: chr "F" "M" "F" "M" ...
> data$gender <- as.factor(gender)
> str(data)
'data.frame': 5 obs. of 2 variables:
 $ idx : num 1 2 3 4 5
 $ gender: Factor w/ 2 levels "F","M": 1 2 1 2 1
> levels(data$gender) <- c("female","male")
> str(data)
'data.frame': 5 obs. of 2 variables:
 $ idx : num 1 2 3 4 5
 $ gender: Factor w/ 2 levels "female","male": 1 2 1 2 1
>
```

#산술평균 구하기

```

153
154 #group by 통해서 산술평균을 구하기
155 height <- c(180, 165, 172, 165, 177, 162, 181, 175, 190 )
156 gender <- c("M","F","M","F","M","F","M","F","M")
157
158 height_gender <- data.frame(height, gender)
159 height_gender
160
161 # aggregate
162 #성별로 키의 평균을 구한다면?
163
164 ?aggregate
165 aggregate(height_gender$height, list(height_gender$gender), mean)
166

```

148:5 (Top Level) <

Console Terminal x Jobs x

~/ ➡

```

3 172 M
4 165 F
5 177 M
6 162 F
7 181 M
8 175 F
9 190 M
>
> # aggregate
> #성별로 키의 평균을 구한다면?
>
> ?aggregate
> aggregate(height_gender$height, list(height_gender$gender), mean)
Group.1 x
1 F 166.75
2 M 180.00
>

```

aggregate(x, by, fun) - by를 기준으로 fun 함수를 사용해 x를 구한다.

```

163 # aggregate(x,by,fun) - by를 기준으로 fun 함수를 사용해 x를 구한다.
164
165 #성별로 키의 평균을 구한다면?
166 ?aggregate
167 aggregate(height_gender$height, list(height_gender$gender), mean)
168
169
170 #예제 데이터 mtcars
171 mtcars
172 str(mtcars)
173 head(mtcars)
174 mtcars[,6]
175
176
177 # cyl 컬럼을 기준으로 나머지 컬럼의 평균 값 구하기
178 aggregate(mtcars, list(cylStandard = mtcars$cyl), mean)
179
180
181 # disp 컬럼이 120 이상인 조건 추가
182 aggregate(mtcars, list(cylStandard = mtcars$cyl, dispHigh = mtcars[, 'disp'] > 120), mean)
183
184 # cyl 컬럼을 기준으로 wt 컬럼의 평균만 구하기
185
186 aggregate(mtcars$wt, list(cyl = mtcars$cyl), mean)
187 aggregate(mtcars[,6], list(cyl = mtcars$cyl), mean)
188 aggregate(wt~cyl, data = mtcars, mean) #formula
189
190
191 # carb , gear 컬럼 두가지를 기준으로 wt 구하기
192 aggregate(wt ~ carb + gear, data = mtcars, mean)
193
194
195 # gear 기준으로 disp, wt 평균 구하기(컬럼을 cbind로 합친다.)
196 aggregate( cbind(disp,wt) ~ gear, data = mtcars, mean)
197

```

```

> aggregate(height_genderheight, list(height_gendergender), mean)
  Group.1      x
1      F 166.75
2      W 100.00
> # cyl 컬럼을 기준으로 나머지 컬럼의 평균 값 구하기
> aggregate(mtcars, list(cylStandard = mtcars$cyl), mean)
  cylStandard      mpg      cyl      disp      hp      drat      wt      qsec      vs      am      gear      carb
1      4 26.66354      4 185.1364 82.63636 4.070000 2.285727 19.13727 0.9090000 0.7272727 4.0909000 1.545455
2      6 19.74286      6 183.3143 122.28571 3.585714 3.117143 17.97714 0.5714286 0.4285714 3.857143 3.428571
3      8 15.10000      8 353.1000 289.21429 3.229286 3.999214 16.77214 0.0000000 0.1428571 3.285714 3.500000
> # disp 컬럼이 120 이상인 조건 추가
> aggregate(mtcars, list(cylStandard = mtcars$cyl, dispHigh = mtcars[, 'disp'] > 120), mean)
  cylStandard dispHigh      mpg      cyl      disp      hp      drat      wt      qsec      vs      am      gear      carb
1      4 FALSE 29.53333      4 84.6000 75.83333 4.155000 1.903000 18.71667 1.0000000 1.0000000 4.166667 1.333333
2      4 TRUE 23.22000      4 129.7000 98.80000 3.970000 2.745000 19.64200 0.8000000 0.4000000 4.000000 1.800000
3      6 TRUE 19.74286      6 183.3143 122.28571 3.585714 3.117143 17.97714 0.5714286 0.4285714 3.857143 3.428571
4      8 TRUE 15.10000      8 353.1000 289.21429 3.229286 3.999214 16.77214 0.0000000 0.1428571 3.285714 3.500000
> aggregate(wt~cyl, data = mtcars, mean) #formula
  cyl      wt
1      4 2.285727
2      6 3.117143
3      8 3.999214
> # carb, gear 컬럼 두가지를 기준으로 wt 구하기
> aggregate(wt~carb + gear, data = mtcars, mean)
  carb gear      wt
1      1      3 3.046667
2      2      3 3.560000
3      3      3 3.860000
4      4      3 4.685000
5      1      4 2.072500
6      2      4 2.683750
7      4      4 3.093750
8      2      5 1.826500
9      4      5 3.170000
10     6      5 2.770000
11     8      5 3.570000
> # carb, gear 컬럼 기준으로 disp, wt 평균
> aggregate( cbind(disp,wt) ~carb + gear, data = mtcars, mean)
  carb gear      disp      wt
1      1      3 201.0333 3.046667
2      2      3 345.5000 3.560000
3      3      3 275.8000 3.860000
4      4      3 415.4000 4.685000
5      1      4 84.2000 2.072500
6      2      4 121.0500 2.683750
7      4      4 163.0000 3.093750
8      2      5 107.7000 1.826500
9      4      5 351.0000 3.170000
10     6      5 145.0000 2.770000
11     8      5 301.0000 3.570000

```

#tapply - 데이터를 색인에 따라 그룹
을 한 후 함수에 따른 결과값을 내는
함수

```
206
207 #tapply(데이터, 색인(그룹), 함수)
208
209 tapply(1:10, rep(1,10), sum)
210 tapply(1:10, 1:10 %%2==0, sum)
211
212 class(tapply(1:10, 1:10 %%2==0, sum))
213
214
215 #iris에서 종별로 Sepal.Length 평균
216 #tapply()
201:1 (Top Level) ⌵
```

Console Terminal x Jobs x

~/ ➡

```
> #tapply(데이터, 색인(그룹), 함수)
>
> tapply(1:10, rep(1,10), sum)
1
55
> tapply(1:10, 1:10 %%2==0, sum)
FALSE TRUE
25 30
>
> class(tapply(1:10, 1:10 %%2==0, sum))
[1] "array"
> |
```



'R' 카테고리의 다른 글

[R] R로 만드는 제어문 (if, else if, for) 과 예제 (0) 02:17:47

[R] R에서 사용되는 Data.frame 과 Fa ctor 에 사용되는 다양한 함수 (0) 02:03:11

[R] R에 사용되는 배열(array)과 리스트(list)의 개념 및 사용되는 함수 (0) 2020.07.23

[R] R에 사용되는 행렬(matrix)의 개념 및 사용되는 함수 (0) 2020.07.23

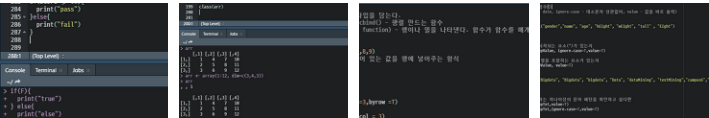
[R] R에서 사용되는 정규표현식(Regex) 표현 방법과 함수를 통한 사용 예제 (0) 2020.07.22

[R] R에 사용되는 벡터(matrix)의 개념 및 사용되는 함수(출력,인덱싱,길이반환,문자열비교 등등) (0) 2020.07.22

태그

- aggregate
- cbind
- colnames
- data.frame
- rbind
- rownames
- tapply
- with
- Within
- 데이터프레임

관련글



[R] R로 만... [R] R에 사... [R] R에 사... [R] R에서 ...

댓글 0

