oracle/R 프로그래밍

R로 데이터를 원하는 모양으로 변형하기 ① (기본함수)

#moon 2014.11.28 12:54

함수	의미
aggregate( )	다양한 함수를 사용하여 계산 결과를 출력
apply( )	
cor( )	상관함수
cumsum( )	설정된 지점까지의 누적합
cumprom( )	설정된 지점까지의 누적곱
diff( )	차이나는 부분을 찾아냄
length( )	요소갯수를 구해서 출력함
max( )	최대값 출력
min( )	최소값 출력
mean( )	평균값 출력
median( )	가운데값 출력
order( )	각 요소의 원래 위치
prod( )	누적곱을 출력
range( )	범위값 출력
rank( )	각 요소의 순위를 출력
rev( )	요소의 역순을 출력
sd( )	표준편차 출력
sort( )	정렬결과 출력
sum( )	총 합계 출력
summary( )	요약 통계량 출력
sweep( )	
tapply( )	벡터에서 주어진 함수연산 수행
var( )	분산값 출력

접기

```
> vec1 <- c(1,2,3,4,5)
> vec2 <- c('a','b','c','d','e')
>
> max(vec1)
[1] 5
> max(vec2)
[1] "e"
> min(vec1)
[1] 1
> min(vec2)
[1] "a"
>
> mean(vec1)
[1] 3
> mean(vec2)
[1] NA
경고메시지:
In mean.default(vec2) :
 인자가 수치형 또는 논리형이 아니므로 NA를 반환합니다

> # 표준편차
> sd(vec1)
[1] 1.581139

> # 합계
> sum(vec1)
[1] 15

> # 분산
> var(vec1)
[1] 2.5
```

접기

2. aggregate 함수 - Dataframe 대상

- 분석할 데이터 형태가 데이터 프레임 형태일 경우 사용하는 함수

- 문법 : aggreate(계산될컬럼~기준될컬럼, 데이터, 함수)

- 기준될 컬럼이 여러개일 경우에는 + 사용 ex) aggregate(Sales~Fruit+Year,Fruits,max)

```
> install.packages("googleVis")
Installing package into ‘C:/Users/stu/Documents/R/win-library/3.1’
(as ‘lib’ is unspecified)
also installing the dependency ‘RJSONIO’

URL 'http://cran.nexr.com/bin/windows/contrib/3.1/RJSONIO_1.3-0.zip'을 시도합니다
Content type 'application/zip' length 1236066 bytes (1.2 Mb)
URL을 열었습니다
downloaded 1.2 Mb

URL 'http://cran.nexr.com/bin/windows/contrib/3.1/googleVis_0.5.6.zip'을 시도합니다
Content type 'application/zip' length 966585 bytes (943 Kb)
URL을 열었습니다
downloaded 943 Kb

패키지 ‘RJSONIO’를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다
패키지 ‘googleVis’를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다
        C:\Users\stu\AppData\Local\Temp\RtmpIBlJIY\downloaded_packages
>
> library(googleVis)
Welcome to googleVis version 0.5.6

Please read the Google API Terms of Use
before you start using the package:
https://developers.google.com/terms/

Note, the plot method of googleVis will by default use
the standard browser to display its output.

See the googleVis package vignettes for more details,
or visit http://github.com/mages/googleVis.

To suppress this message use:
suppressPackageStartupMessages(library(googleVis))
> Fruits
  Fruit Year Location Sales Expenses Profit     Date
1 Apples 2008    West   98       78     20 2008-12-31
```

```
5 Bananas 2008    East   85     76     9 2008-12-31
6 Oranges 2009    East   93     80    13 2009-12-31
7 Bananas 2009    East   94     78    16 2009-12-31
8 Oranges 2010    East   98     91     7 2010-12-31
9 Bananas 2010    East   81     71    10 2010-12-31
>
> # 년도별로 salse된 양을 sum하기
> aggregate(Sales~year,Fruits,sum)
다음에 오류가 있습니다eval(expr, envir, enclos) : 객체 'year'를 찾을 수 없습니다
> aggregate(Sales~Year,Fruits,sum)
  Year Sales
1 2008   279
2 2009   298
3 2010   268
>
> # Fruit별로 Sales된 수량을 sum하기
> aggregate(Sales~Fruit,Fruits,sum)
    Fruit Sales
1  Apples   298
2 Bananas   260
3 Oranges   287
>
> # 과일별로 가장 많이 Sales된 수량 조회
> aggregate(Sales~Fruit,Fruits,max)
    Fruit Sales
1  Apples   111
2 Bananas    94
3 Oranges    98
>
> # 과일별, 년도별 최대 판매량
> aggregate(Sales~Fruit+Year,Fruits,max)
    Fruit Year Sales
1  Apples 2008    98
2 Bananas 2008    85
3 Oranges 2008    96
4  Apples 2009   111
5 Bananas 2009    94
6 Oranges 2009    93
7  Apples 2010    89
```

3. apply 함수로 분석하기 - Matrix 대상

- R에서 아주 많이 사용되는 함수이며 여러가지 변형도 많음
- 문법 : apply(대상,행/열,적용함수)
- 행/열 -> 1이면 행, 2이면 열기준으로 적용
- 행렬(Matrix)일 경우에 유용하게 사용됨 -> 행이나 열을 대상으로 작업하기때문에

접기

```
> mat1 <- matrix(c(1,2,3,4,5,6), nrow=2, byrow=T)
> mat1
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
>
> # 각행의 합계구하기
> apply(mat1,1,sum)
[1]  6 15
>
> # 각열의 합계 구가힉
> apply(mat1,2,sum)
[1] 5 7 9
>
> # 2열과 3열의 최대값 구하기
> apply(mat1[,c(2,3)],2,max)
[1] 5 6
```

접기

4. lapply / sapply 함수

- apply함수의 변형
- 문법 : lapply/sapply(대상, 적용함수)

접기

```
> # list형의 데이터에 적용한 예
> list1 <- list(Fruits$Sales)
> list1
[[1]]
[1] 98 111 89 96 85 93 94 98 81


> list2 <- list(Fruits$Profit)
```

```
> lapply(c(list1,list2),max)
[[1]]
[1] 111

[[2]]
[1] 32

> sapply(c(list1,list2),max)
[1] 111  32

> # dataframe에 적용한 예
> Fruits
   Fruit Year Location Sales Expenses Profit      Date
1  Apples 2008    West   98       78     20 2008-12-31
2  Apples 2009    West  111       79     32 2009-12-31
3  Apples 2010    West   89       76     13 2010-12-31
4 Oranges 2008    East   96       81     15 2008-12-31
5 Bananas 2008    East   85       76      9 2008-12-31
6 Oranges 2009    East   93       80     13 2009-12-31
7 Bananas 2009    East   94       78     16 2009-12-31
8 Oranges 2010    East   98       91      7 2010-12-31
9 Bananas 2010    East   81       71     10 2010-12-31
>
> lapply(Fruits[,c(4,5)],max)
$Sales
[1] 111

$Expenses
[1] 91

> sapply(Fruits[,c(4,5)],max)
  Sales Expenses
    111       91
```

접기

5. tapply / mapply 함수

tapply( )

- 그룹별 처리를 위한 apply함수
- 문법 : **tapply(출력값, 기준컬럼, 적용함수)**

mapply( )

- 데이터프레임이 아닌 벡터나 리스트 형태의 데이터를 마치 데이터 프레임처럼 연산을 해주는 함수
- 문법 : **mapply(함수, 벡터1, 벡터2, 벡터3 ...)**

접기

```
2  Apples 2009    West  111     79    32 2009-12-31
3  Apples 2010    West   89     76    13 2010-12-31
4 Oranges 2008    East   96     81    15 2008-12-31
5 Bananas 2008    East   85     76     9 2008-12-31
6 Oranges 2009    East   93     80    13 2009-12-31
7 Bananas 2009    East   94     78    16 2009-12-31
8 Oranges 2010    East   98     91     7 2010-12-31
9 Bananas 2010    East   81     71    10 2010-12-31
>
> tapply(Sales,Fruit,sum)
다음에 오류가 있습니다tapply(Sales, Fruit, sum) : 객체 'Fruit'를 찾을 수 없습니다
>
> # attach(데이터) : 컬럼명을 변수명처럼 처리
> attach(Fruits)
> tapply(Sales,Fruit,sum)
 Apples Bananas Oranges
   298    260    287

> # mapply 함수사용하기, 벡터형태의 데이터 생성
> vec1 <- c(1,2,3,4,5)
> vec2 <- c(10,20,30,40,50)
> vec3 <- c(100,200,300,400,500)
>
> mapply(sum,vec1,vec2,vec3)
[1] 111 222 333 444 555
```

접기

6. sweep 함수 - 한꺼번에 차이 구하기

- 벡터, 매트릭스, 배열, 데이터프레임으로 구성된 여러 데이터들에 동일한 기준을 적용시켜 차이나는 부분을 한꺼번에 보여주는 함수

접기

```
> mat1
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
>
> a <- c(1,1,1)
> sweep(mat1,2,a)
```

#moon
all about moon

접기

7. length 함수 - 요소의 개수나 줄수 파악하기

- 요소의 개수나 줄수 파악하는 함수

접기

```
> a <- c(1,2,3,4,5)
> length(a)
[1] 5
>
> # 데이터프레임의 경우는 라벨수를 출력
> Fruits
   Fruit Year Location Sales Expenses Profit     Date
1  Apples 2008    West   98     78    20 2008-12-31
2  Apples 2009    West  111     79    32 2009-12-31
3  Apples 2010    West   89     76    13 2010-12-31
4 Oranges 2008    East   96     81    15 2008-12-31
5 Bananas 2008    East   85     76     9 2008-12-31
6 Oranges 2009    East   93     80    13 2009-12-31
7 Bananas 2009    East   94     78    16 2009-12-31
8 Oranges 2010    East   98     91     7 2010-12-31
9 Bananas 2010    East   81     71    10 2010-12-31
> length(Fruits)
[1] 7
```

접기

'oracle > R 프로그래밍' 카테고리의 다른 글

R로 데이터를 원하는 모양으로 변형하기 ③ (사용... (0)
R로 데이터를 원하는 모양으로 변형하기 ① (기본... (0)
R 기초 문법 ② (벡터, 행렬, list, 데이터 프레임) (0)
R로 데이터를 원하는 모양으로 변형하기 ② (숫자... (0)
R에서 데이터 불러오기/저장하기 (0)
R 기초 문법 ① (R특징, 데이터형, 변수) (1)

#moon
all about moon

CATEGORY

이름 비밀번호 비밀글 입력

1 2 3 4 5 6 7 8 9 10 11

공지사항

최근에 올라온 글

ㅇㅇㅇ
Redis Sentinel
MIME-Type List
git command

최근에 달린 댓글

as.data 포맷어떻게하죠?
내용 중 결합인덱스 관련해 오..
좋은 정보 잘보고 갑니다
좋은 글 잘 보고 갑니다 ^^ 감..

Total

128,898

Today 0
Yesterday 68

링크

조대협의 블로그
MySQL DBA를 위한
::개발자가 사는 세상::
오라클 스터디
일상다반사
오라클 취업반 올해 3기
'진지'드세용
에너쓰오라클
유유a

TAG

more

« 2019/05 »

일	월	화	수	목	금	토
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

글 보관함

2016/03 (1)
2016/02 (1)
2015/12 (1)
2015/11 (1)