plyr in R

# 简介

本文将着重介绍一下同一个作者开发的plyr包，该包与dplyr包既有相似的地方，也有不同的地方。plyr包最让我兴奋的地方是可以方便的实现数据结构之间的转换，具体见下文的详细说明。

# 函数介绍及例子

## a\*ply函数格式 aaply(.data = ,.margins = ,.fun = ,...,.progress = 'none',.inform = FALSE) adply(.data = ,.margins = ,.fun = ,...,.progress = 'none',.inform = FALSE) alply(.data = ,.margins = ,.fun = ,...,.progress = 'none',.inform = FALSE) a\_ply(.data = ,.margins = ,.fun = ,...,.progress = 'none',.inform = FALSE) .data可以是数组也可以是矩阵； .margins指定要分析的数组或矩阵的维度，即行维(.margins = 1)还是列维(.margins = 2)； .fun为行维或列维指定需要处理的函数，可以是R自带的函数，如sum()，mean()等，也可以是自定义函数； ...为指定函数的其他参数； .progress指定以什么样的方式展示程序运行的进度，默认不显示进度，还可以选择"text"(文本进度条), "tk"(tk进度条), 和"win"(Windows系统自带的进度条)； .inform是否指定报错信息，默认不指定，因为设为TRUE的话，将降低程序的执行效率，但该参数对Bug处理是有帮助的。

## 例子：

library(plyr)  
a<-array(data=1:500000,dim=c(100000,5))  
test1<-aaply(.data = a,.margins = 1,.fun = mean, .progress = 'none') #对每一行求均值，不显示进度条  
head(test1)

## 1 2 3 4 5 6   
## 200001 200002 200003 200004 200005 200006

test2<-adply(.data = a,.margins = 1,.fun = sd, .progress = 'text') #对每一行求标准差，以文本形式显示进度条

##   
 |   
 | | 0%  
 |   
 | | 1%  
 |   
 |= | 1%  
 |   
 |= | 2%  
 |   
 |== | 2%  
 |   
 |== | 3%  
 |   
 |== | 4%  
 |   
 |=== | 4%  
 |   
 |=== | 5%  
 |   
 |==== | 5%  
 |   
 |==== | 6%  
 |   
 |==== | 7%  
 |   
 |===== | 7%  
 |   
 |===== | 8%  
 |   
 |====== | 8%  
 |   
 |====== | 9%  
 |   
 |====== | 10%  
 |   
 |======= | 10%  
 |   
 |======= | 11%  
 |   
 |======= | 12%  
 |   
 |======== | 12%  
 |   
 |======== | 13%  
 |   
 |========= | 13%  
 |   
 |========= | 14%  
 |   
 |========= | 15%  
 |   
 |========== | 15%  
 |   
 |========== | 16%  
 |   
 |=========== | 16%  
 |   
 |=========== | 17%  
 |   
 |=========== | 18%  
 |   
 |============ | 18%  
 |   
 |============ | 19%  
 |   
 |============= | 19%  
 |   
 |============= | 20%  
 |   
 |============= | 21%  
 |   
 |============== | 21%  
 |   
 |============== | 22%  
 |   
 |=============== | 22%  
 |   
 |=============== | 23%  
 |   
 |=============== | 24%  
 |   
 |================ | 24%  
 |   
 |================ | 25%  
 |   
 |================= | 25%  
 |   
 |================= | 26%  
 |   
 |================= | 27%  
 |   
 |================== | 27%  
 |   
 |================== | 28%  
 |   
 |=================== | 28%  
 |   
 |=================== | 29%  
 |   
 |=================== | 30%  
 |   
 |==================== | 30%  
 |   
 |==================== | 31%  
 |   
 |==================== | 32%  
 |   
 |===================== | 32%  
 |   
 |===================== | 33%  
 |   
 |====================== | 33%  
 |   
 |====================== | 34%  
 |   
 |====================== | 35%  
 |   
 |======================= | 35%  
 |   
 |======================= | 36%  
 |   
 |======================== | 36%  
 |   
 |======================== | 37%  
 |   
 |======================== | 38%  
 |   
 |========================= | 38%  
 |   
 |========================= | 39%  
 |   
 |========================== | 39%  
 |   
 |========================== | 40%  
 |   
 |========================== | 41%  
 |   
 |=========================== | 41%  
 |   
 |=========================== | 42%  
 |   
 |============================ | 42%  
 |   
 |============================ | 43%  
 |   
 |============================ | 44%  
 |   
 |============================= | 44%  
 |   
 |============================= | 45%  
 |   
 |============================== | 45%  
 |   
 |============================== | 46%  
 |   
 |============================== | 47%  
 |   
 |=============================== | 47%  
 |   
 |=============================== | 48%  
 |   
 |================================ | 48%  
 |   
 |================================ | 49%  
 |   
 |================================ | 50%  
 |   
 |================================= | 50%  
 |   
 |================================= | 51%  
 |   
 |================================= | 52%  
 |   
 |================================== | 52%  
 |   
 |================================== | 53%  
 |   
 |=================================== | 53%  
 |   
 |=================================== | 54%  
 |   
 |=================================== | 55%  
 |   
 |==================================== | 55%  
 |   
 |==================================== | 56%  
 |   
 |===================================== | 56%  
 |   
 |===================================== | 57%  
 |   
 |===================================== | 58%  
 |   
 |====================================== | 58%  
 |   
 |====================================== | 59%  
 |   
 |======================================= | 59%  
 |   
 |======================================= | 60%  
 |   
 |======================================= | 61%  
 |   
 |======================================== | 61%  
 |   
 |======================================== | 62%  
 |   
 |========================================= | 62%  
 |   
 |========================================= | 63%  
 |   
 |========================================= | 64%  
 |   
 |========================================== | 64%  
 |   
 |========================================== | 65%  
 |   
 |=========================================== | 65%  
 |   
 |=========================================== | 66%  
 |   
 |=========================================== | 67%  
 |   
 |============================================ | 67%  
 |   
 |============================================ | 68%  
 |   
 |============================================= | 68%  
 |   
 |============================================= | 69%  
 |   
 |============================================= | 70%  
 |   
 |============================================== | 70%  
 |   
 |============================================== | 71%  
 |   
 |============================================== | 72%  
 |   
 |=============================================== | 72%  
 |   
 |=============================================== | 73%  
 |   
 |================================================ | 73%  
 |   
 |================================================ | 74%  
 |   
 |================================================ | 75%  
 |   
 |================================================= | 75%  
 |   
 |================================================= | 76%  
 |   
 |================================================== | 76%  
 |   
 |================================================== | 77%  
 |   
 |================================================== | 78%  
 |   
 |=================================================== | 78%  
 |   
 |=================================================== | 79%  
 |   
 |==================================================== | 79%  
 |   
 |==================================================== | 80%  
 |   
 |==================================================== | 81%  
 |   
 |===================================================== | 81%  
 |   
 |===================================================== | 82%  
 |   
 |====================================================== | 82%  
 |   
 |====================================================== | 83%  
 |   
 |====================================================== | 84%  
 |   
 |======================================================= | 84%  
 |   
 |======================================================= | 85%  
 |   
 |======================================================== | 85%  
 |   
 |======================================================== | 86%  
 |   
 |======================================================== | 87%  
 |   
 |========================================================= | 87%  
 |   
 |========================================================= | 88%  
 |   
 |========================================================== | 88%  
 |   
 |========================================================== | 89%  
 |   
 |========================================================== | 90%  
 |   
 |=========================================================== | 90%  
 |   
 |=========================================================== | 91%  
 |   
 |=========================================================== | 92%  
 |   
 |============================================================ | 92%  
 |   
 |============================================================ | 93%  
 |   
 |============================================================= | 93%  
 |   
 |============================================================= | 94%  
 |   
 |============================================================= | 95%  
 |   
 |============================================================== | 95%  
 |   
 |============================================================== | 96%  
 |   
 |=============================================================== | 96%  
 |   
 |=============================================================== | 97%  
 |   
 |=============================================================== | 98%  
 |   
 |================================================================ | 98%  
 |   
 |================================================================ | 99%  
 |   
 |=================================================================| 99%  
 |   
 |=================================================================| 100%

head(test2)

## X1 V1  
## 1 1 158113.9  
## 2 2 158113.9  
## 3 3 158113.9  
## 4 4 158113.9  
## 5 5 158113.9  
## 6 6 158113.9

a2<-array(rnorm(10000000), dim = c(1000,10000))  
test3<-alply(.data = a2,.margins = 2,.fun = sum, .progress = 'tk') #对每一列求和，以tk形式显示进度条  
a3<-array(rnorm(10000000), dim = c(100,100000))  
test4<-a\_ply(.data = a3,.margins = 2,.fun = max, .progress = 'win') #对每一列求最大值，以Windows自带进度条显示进度

## d*ply函数格式 daply(.data = ,.variables = ,.fun = ,...,.progress = 'none',.inform = FALSE) ddply(.data = ,.variables = ,.fun = ,...,.progress = 'none',.inform = FALSE) dlply(.data = ,.variables = ,.fun = ,...,.progress = 'none',.inform = FALSE) d\_ply(.data = ,.variables = ,.fun = ,...,.progress = 'none',.inform = FALSE) .data指定为数据框数据； .variables指定数据框中的分组变量； .fun基于分组变量，可对数据框中的其余变量指定某种函数，可以是R自带的函数，如sum()，mean()等，也可以是自定义函数，类似于聚合分析； .progress和.inform与a*ply函数参数一致。

## 例子：

fun<-function(data) apply(data,2,mean) #构建自定义函数  
daply(.data = iris[,1:4],.variables = .(iris$Species),.fun=fun)

##   
## iris$Species Sepal.Length Sepal.Width Petal.Length Petal.Width  
## setosa 5.006 3.428 1.462 0.246  
## versicolor 5.936 2.770 4.260 1.326  
## virginica 6.588 2.974 5.552 2.026

ddply(.data = iris[,1:4],.variables = .(iris$Species),.fun=fun)

## iris$Species Sepal.Length Sepal.Width Petal.Length Petal.Width  
## 1 setosa 5.006 3.428 1.462 0.246  
## 2 versicolor 5.936 2.770 4.260 1.326  
## 3 virginica 6.588 2.974 5.552 2.026

dlply(.data = iris[,1:4],.variables = .(iris$Species),.fun=fun)

## $setosa  
## Sepal.Length Sepal.Width Petal.Length Petal.Width   
## 5.006 3.428 1.462 0.246   
##   
## $versicolor  
## Sepal.Length Sepal.Width Petal.Length Petal.Width   
## 5.936 2.770 4.260 1.326   
##   
## $virginica  
## Sepal.Length Sepal.Width Petal.Length Petal.Width   
## 6.588 2.974 5.552 2.026   
##   
## attr(,"split\_type")  
## [1] "data.frame"  
## attr(,"split\_labels")  
## iris$Species  
## 1 setosa  
## 2 versicolor  
## 3 virginica

d\_ply(.data = iris[,1:4],.variables = .(iris$Species),.fun=fun)

## l*ply函数格式 laply(.data = ,.fun = ,...,.progress = 'none',.inform = FALSE) ldply(.data = ,.fun = ,...,.progress = 'none',.inform = FALSE) llply(.data = ,.fun = ,...,.progress = 'none',.inform = FALSE) l\_ply(.data = ,.fun = ,...,.progress = 'none',.inform = FALSE) .data可以指定为列表数据，也可以指定为向量数据； 其余参数与a*ply()函数和d\*lpy()函数参数一致。

## 例子：

x1<-1:100  
x2<-seq(from = 100,to = 1000 ,by = 2)  
x3<-runif(150,min = 10,max = 100)  
##列表由向量构成  
l1<-list(x1 = x1,x2 = x2,x3 = x3)  
laply(.data = l1,.fun = mean)

## [1] 50.50000 550.00000 56.07177

ldply(.data = l1,.fun = summary)

## .id Min. 1st Qu. Median Mean 3rd Qu. Max.  
## 1 x1 1.0 25.75 50.50 50.50 75.25 100.00  
## 2 x2 100.0 325.00 550.00 550.00 775.00 1000.00  
## 3 x3 10.1 35.60 57.53 56.07 75.05 99.86

llply(.data = l1,.fun = quantile)

## $x1  
## 0% 25% 50% 75% 100%   
## 1.00 25.75 50.50 75.25 100.00   
##   
## $x2  
## 0% 25% 50% 75% 100%   
## 100 325 550 775 1000   
##   
## $x3  
## 0% 25% 50% 75% 100%   
## 10.10031 35.59940 57.52962 75.05371 99.86129

l\_ply(.data = l1,.fun = summary)

## 构建数据框d11

y11<-rnorm(n = 100,mean = 10,sd = 5)  
y12<-rt(n = 100,df = 3)  
y13<-rf(n = 100,df1 = 2,df2 = 3)  
y14<-factor(x = c('low','potential','high'),ordered = T)  
y15<-sample(y14,size = 100,replace = TRUE)  
d11<-data.frame(y1 = y11,y2 = y12,y3 = y13,y5 = y15)  
head(d11)

## y1 y2 y3 y5  
## 1 7.426917 2.2134761 9.1005128 high  
## 2 6.087976 1.5652724 0.4363494 low  
## 3 9.606745 -1.6864277 4.0849724 high  
## 4 7.218128 0.8596896 1.7647805 potential  
## 5 8.118099 2.0234735 0.6066718 potential  
## 6 11.806497 -1.1614848 1.0798862 low

## 构建数据框d21

y21<-1:100  
y22<-seq(from = 1,to = 2,length = 100)  
y23<-rchisq(n = 100,df = 8)  
y24<-factor(x = c('A','B','C','D'),ordered = T)  
y25<-sample(y24,size = 100,replace = TRUE)  
d21<-data.frame(y21 = y21,y22 = y22,y23 = y23,y25 = y25)  
head(d21)

## y21 y22 y23 y25  
## 1 1 1.000000 2.161869 B  
## 2 2 1.010101 3.990570 D  
## 3 3 1.020202 5.100406 B  
## 4 4 1.030303 14.014288 A  
## 5 5 1.040404 4.965534 B  
## 6 6 1.050505 18.131077 D

## 列表由数据框组成

l2<-list(first = d11,second = d21)  
library(psych)  
fun<-function(data) describeBy(data[,1:3],group = data[,4])  
llply(.data = l2,.fun = fun,.progress = 'none')

## $first  
## $high  
## vars n mean sd median trimmed mad min max range skew  
## y1 1 32 12.63 5.98 13.29 12.11 6.35 5.05 31.18 26.14 0.81  
## y2 2 32 -0.47 2.58 -0.40 -0.15 0.98 -13.01 2.47 15.48 -3.47  
## y3 3 32 4.41 8.18 0.83 2.40 0.91 0.00 35.67 35.66 2.43  
## kurtosis se  
## y1 0.71 1.06  
## y2 14.58 0.46  
## y3 5.50 1.45  
##   
## $low  
## vars n mean sd median trimmed mad min max range skew kurtosis  
## y1 1 37 11.02 4.89 11.51 11.00 3.55 0.56 21.13 20.57 0.01 -0.55  
## y2 2 37 -0.20 1.22 -0.18 -0.20 0.84 -3.97 3.17 7.14 -0.14 1.77  
## y3 3 37 2.39 4.54 0.96 1.45 0.77 0.12 26.10 25.98 4.02 17.49  
## se  
## y1 0.80  
## y2 0.20  
## y3 0.75  
##   
## $potential  
## vars n mean sd median trimmed mad min max range skew kurtosis  
## y1 1 31 10.18 4.54 10.16 10.13 4.49 1.56 20.99 19.43 0.17 -0.39  
## y2 2 31 0.17 1.02 0.38 0.21 0.98 -2.06 2.02 4.08 -0.40 -0.56  
## y3 3 31 3.48 5.58 0.82 2.32 1.02 0.06 26.76 26.69 2.54 7.17  
## se  
## y1 0.82  
## y2 0.18  
## y3 1.00  
##   
## attr(,"call")  
## by.data.frame(data = x, INDICES = group, FUN = describe, type = type)  
##   
## $second  
## $A  
## vars n mean sd median trimmed mad min max range skew  
## y21 1 22 49.18 27.56 52.50 48.78 31.88 4.00 97.00 93.00 0.03  
## y22 2 22 1.49 0.28 1.52 1.48 0.32 1.03 1.97 0.94 0.03  
## y23 3 22 8.39 4.92 7.74 8.18 5.93 1.52 18.48 16.97 0.35  
## kurtosis se  
## y21 -1.29 5.88  
## y22 -1.29 0.06  
## y23 -1.16 1.05  
##   
## $B  
## vars n mean sd median trimmed mad min max range skew  
## y21 1 28 46.68 31.64 49.50 46.54 45.22 1.00 95.00 94.00 0.03  
## y22 2 28 1.46 0.32 1.49 1.46 0.46 1.00 1.95 0.95 0.03  
## y23 3 28 8.60 4.41 8.78 8.20 4.41 2.16 23.93 21.77 1.27  
## kurtosis se  
## y21 -1.55 5.98  
## y22 -1.55 0.06  
## y23 2.83 0.83  
##   
## $C  
## vars n mean sd median trimmed mad min max range skew  
## y21 1 25 52.48 27.90 47.00 51.90 37.06 8.00 99.00 91.00 0.22  
## y22 2 25 1.52 0.28 1.46 1.51 0.37 1.07 1.99 0.92 0.22  
## y23 3 25 7.62 4.00 6.56 7.54 3.64 0.87 16.08 15.21 0.29  
## kurtosis se  
## y21 -1.37 5.58  
## y22 -1.37 0.06  
## y23 -0.84 0.80  
##   
## $D  
## vars n mean sd median trimmed mad min max range skew  
## y21 1 25 53.96 29.46 53.00 54.52 32.62 2.00 100.00 98.00 -0.19  
## y22 2 25 1.53 0.30 1.53 1.54 0.33 1.01 2.00 0.99 -0.19  
## y23 3 25 10.30 5.59 8.65 9.78 4.78 3.26 22.74 19.48 0.80  
## kurtosis se  
## y21 -1.16 5.89  
## y22 -1.16 0.06  
## y23 -0.47 1.12  
##   
## attr(,"call")  
## by.data.frame(data = x, INDICES = group, FUN = describe, type = type)

llply(.data = l2,.fun = fun,.progress = 'text')

##   
 |   
 | | 0%  
 |   
 |================================ | 50%  
 |   
 |=================================================================| 100%

## $first  
## $high  
## vars n mean sd median trimmed mad min max range skew  
## y1 1 32 12.63 5.98 13.29 12.11 6.35 5.05 31.18 26.14 0.81  
## y2 2 32 -0.47 2.58 -0.40 -0.15 0.98 -13.01 2.47 15.48 -3.47  
## y3 3 32 4.41 8.18 0.83 2.40 0.91 0.00 35.67 35.66 2.43  
## kurtosis se  
## y1 0.71 1.06  
## y2 14.58 0.46  
## y3 5.50 1.45  
##   
## $low  
## vars n mean sd median trimmed mad min max range skew kurtosis  
## y1 1 37 11.02 4.89 11.51 11.00 3.55 0.56 21.13 20.57 0.01 -0.55  
## y2 2 37 -0.20 1.22 -0.18 -0.20 0.84 -3.97 3.17 7.14 -0.14 1.77  
## y3 3 37 2.39 4.54 0.96 1.45 0.77 0.12 26.10 25.98 4.02 17.49  
## se  
## y1 0.80  
## y2 0.20  
## y3 0.75  
##   
## $potential  
## vars n mean sd median trimmed mad min max range skew kurtosis  
## y1 1 31 10.18 4.54 10.16 10.13 4.49 1.56 20.99 19.43 0.17 -0.39  
## y2 2 31 0.17 1.02 0.38 0.21 0.98 -2.06 2.02 4.08 -0.40 -0.56  
## y3 3 31 3.48 5.58 0.82 2.32 1.02 0.06 26.76 26.69 2.54 7.17  
## se  
## y1 0.82  
## y2 0.18  
## y3 1.00  
##   
## attr(,"call")  
## by.data.frame(data = x, INDICES = group, FUN = describe, type = type)  
##   
## $second  
## $A  
## vars n mean sd median trimmed mad min max range skew  
## y21 1 22 49.18 27.56 52.50 48.78 31.88 4.00 97.00 93.00 0.03  
## y22 2 22 1.49 0.28 1.52 1.48 0.32 1.03 1.97 0.94 0.03  
## y23 3 22 8.39 4.92 7.74 8.18 5.93 1.52 18.48 16.97 0.35  
## kurtosis se  
## y21 -1.29 5.88  
## y22 -1.29 0.06  
## y23 -1.16 1.05  
##   
## $B  
## vars n mean sd median trimmed mad min max range skew  
## y21 1 28 46.68 31.64 49.50 46.54 45.22 1.00 95.00 94.00 0.03  
## y22 2 28 1.46 0.32 1.49 1.46 0.46 1.00 1.95 0.95 0.03  
## y23 3 28 8.60 4.41 8.78 8.20 4.41 2.16 23.93 21.77 1.27  
## kurtosis se  
## y21 -1.55 5.98  
## y22 -1.55 0.06  
## y23 2.83 0.83  
##   
## $C  
## vars n mean sd median trimmed mad min max range skew  
## y21 1 25 52.48 27.90 47.00 51.90 37.06 8.00 99.00 91.00 0.22  
## y22 2 25 1.52 0.28 1.46 1.51 0.37 1.07 1.99 0.92 0.22  
## y23 3 25 7.62 4.00 6.56 7.54 3.64 0.87 16.08 15.21 0.29  
## kurtosis se  
## y21 -1.37 5.58  
## y22 -1.37 0.06  
## y23 -0.84 0.80  
##   
## $D  
## vars n mean sd median trimmed mad min max range skew  
## y21 1 25 53.96 29.46 53.00 54.52 32.62 2.00 100.00 98.00 -0.19  
## y22 2 25 1.53 0.30 1.53 1.54 0.33 1.01 2.00 0.99 -0.19  
## y23 3 25 10.30 5.59 8.65 9.78 4.78 3.26 22.74 19.48 0.80  
## kurtosis se  
## y21 -1.16 5.89  
## y22 -1.16 0.06  
## y23 -0.47 1.12  
##   
## attr(,"call")  
## by.data.frame(data = x, INDICES = group, FUN = describe, type = type)

# 总结

plyr包中还有很多其他函数，如rename()、arrange()、join()等函数，其功能与dplyr包中的rename()、arrange()、left\_join()一致，这里不再赘述。这两个包可以相辅相成，互补缺陷，各显优势，所以在数据预处理过程中可以考虑结合使用这两个包，一定能使工作效率得到大幅提升。