R three dimensional arrays [closed]

I have an 3D array - arr

```
dim(arr)
[1] 2000   22    3
```

Let's call 2000 `dim1` , 22 `dim2` , and 3 `dim3` . I would like to do something like this:

```
for(d1 in 1:dim1){
    for(d2 in 1:dim2){
        #compute the mean across the 2000 values of arr[d1,d2]
        m<-mean(arr[d1,d2])
    }
}
```

Any idea?

arrays r

edited Feb 27 '13 at 21:02
Julius
16.6k 6 42 60

asked Feb 27 '13 at 20:58
user1701545
1,203 2 12 30

closed as not a real question by Jack Maney, csgillespie, sebastian-c, Tyler Rinker, plannapus Mar 1 '13 at 8:16

It's difficult to tell what is being asked here. This question is ambiguous, vague, incomplete, overly broad, or rhetorical and cannot be reasonably answered in its current form. For help clarifying this question so that it can be reopened, visit the help center.

If this question can be reworded to fit the rules in the help center, please edit the question.

1 What are you asking? How to make same computations faster or what? – Hemmo Feb 27 '13 at 21:04

3

@Hemmo: he clearly would get errors with the code as it stands. Make the example smaller and then use it for testing. Any reference to the `arr` -object needs to have two commas inside the "[" call. – 42- Feb 27 '13 at 21:05

1 (-1), after looking at the post and the OP's comments, I really can't figure out what the question is. – QkuCeHBH Feb 27 '13 at 23:56

3 Answers

So the way your question is posed is a bit sloppy, but an example of what you might be trying to do is to take the average of each 2000 x 22 array for each of the 3 of them. here is how this would be done:

```
arr = array(1, dim=c(2000,22,3))
dim(arr)

m = NULL
dim3 = 3
for(d3 in 1:dim3){
    m[d3] = mean(arr[,,d3])
}
```

answered Feb 27 '13 at 21:30

user2005253

Thanks. Let me try to further clarify it. Is there a way to convert this array into a list of 3 [22 X 2000] matrices? Then I can loop over the list and compute the mean for each matrix row. Thanks – user1701545 Feb 27 '13 at 21:52

I'm guessing your might want this result:

```
m <- apply(arr, 1:2, mean)
```

Your current code would overwrite m at every inner iteration and you would end up with a single value, and it would also throw a dimension mismatch error when it encountered `arr[d1,d2]` . If

you wanted to use a for-loop starategy, you would needed to define `m` as a dim1 x dim2 matrix and then populate its entries using `m[d1,d2] <- mean(arr[d1, d2, ])` .

Next time if you want to avoid all of those downvotes, why not show some testing with small example object:

```
arr <- array(1:5*4*3, c(5,4,3))
```

edited Feb 27 '13 at 21:39

answered Feb 27 '13 at 21:21

42-

165k 8 145 275

In the following Q&A you find little more code for a comparable situation and answer. It might be helpful to look at that code together with this answer: stackoverflow.com/questions/13475039/… – Jochem Feb 27 '13 at 21:27

My apologies for not being clear. Basically, not having much experience with R 3D arrays, my question is how do I perform an action like computing the mean over all d3 values, for each d2 dimension of each d1 dimension of an array with these dimensions: dim(arr) [1] d3 d2 d1 – user1701545 Feb 27 '13 at 21:35

Never mind, I got it. That's how I would create a list of 3 [4 x 6] matrices:

```
arr = array(1, dim=c(6,4,3));
l<-list(length=dim(arr)[3]);
for (i in 1:dim(arr)[3]){
    l[[i]]<-arr[,,i];
}
```

cheers

answered Feb 27 '13 at 22:21

user1701545

1,203 2 12 30