1. Today I’m gonna talk about apply functions in R

Well in real life the apply function on the job application page might not be really useful, cuz regrettably most of the times it didn’t return any useful results at all. but in R Studio, the family of apply functions are extremely useful and convenient when applied properly.

2. So when you search the keyword apply in RStudio, it returns something like this. What these apply functions do is to applying functions to manipulate slices of data like matrix, array, and vector, Their results can also be useful to generate the data that are useful and suitable for the plots. With these apply functions we can avoid looping in other programming languages like java, and write code that makes more sense in a statistical way. For the time limit of this presentation, I will only cover the ones that are more could frequently used, which are apply, tapply, lapply, and sapply.

3. Ok let’s talk about the grandfather of all apply functions first. The apply function applies a function to every row or every column or every row and column of a MATRIX.

Usually it looks like this: apply(X, MARGIN, FUN, ...)

Where X is an array or a matrix, MARGIN value could be 1, 2 or c(1,2). 1 means applies to every row, 2 means apply to every column, and c(1,2) means apply to every row and column.

FUN is the function we want to apply, this could be mean, max, median, sum, etc..or some of the definitions that you created.

And it returns a vector.

So here it goes

5. Lapply and sapply could be useful if you want to

1. Apply a function to each list element
2. Apply a function to every column in a dataframe

The difference: lapply returns a list, sapply returns a vector. List is a type of vector that could possibly contain different data types like characters and numbers, while vector can only contain one data type.

Lets take them for a spin.

lapply (x,fun). sapply (s,fun)

x is the list or vector(s) you want to have some fun, and fun is the fun.

so if we apply the class function on our classic mtcars dataset with lapply and sapply, you get results of the type of classes of each column in the dataset. The difference is, lapply returns a list and sapply returns a vector, you get the idea.

6. Now here’s the fun part, the tapply. We have encountered quite a bit of tapplies in our labs and hws, as it’s useful for plotting comparisons between groups. So it’s more like a recap. Basically it applies a function or operation on subset of the vector broken down by a given factor variable. What it means is we break the dataset into groups and you wanna process the data by the groups, for example, summing by group or averaging by group.

The simply version of a tapply looks like this:

X is a vector, INDEX is a grouping factor, and fun is a function. The whole thing means apply a function to x grouped by INDEX.

For example, for our classic mtcars dataset. If we wanna see MEAN mpg GROUP BY cyl we could do a little something like this

tapply(art$units.sold,art$store,sum), this returns an array

and you just barplot it and you get a nice barplot like that, and it’s obvious to see the more cyl a car has the lower the mpg is gonna be

7.this returns a matrix. In the matrix, cyl is the row name and gear is the column names.

And we can generate this graph with a pretty simple barplot function and we can see the number of cylinder has more to do with mpg compare to the number of gear

8. When I was doing homework I was confused about the aggregate function and tapply because they are kind of similar.

aggregate is designed to work on multiple columns with one function and return a dataframe with one row for each category, while tapply is designed to work on a single vector with results returned as a matrix or array. There’s no car in the list with 4 gear and 8 cylinder that’s why aggregate doesn’t list it.

For the mean of the mpg grouped by cyl and gears, the results are like these

9. Ok so these are the functions I personally used frequently, but there are more useful functions out there in both core package and 3rd party packages and they can do a lot of things. Overall I had a lot of fun preparing for this presentation as I spent quite a while playing with different apply functions in RStudio, changing different apply functions and their arguments and compare their results and et cetera, and I encourage you to do so too cuz I only presented the most basic forms of some of the apply functions, also a tons of arguments to play with in each functions. It’s the best way to actually learn the differences between all these apply functions and ace them.

So I guess that’s all, thank you guys for listening and good luck with the poster.

What’s the difference between a list and a vector?

lists are "recursive". This means that they can contain values of different types, even other lists

Whereas all the elements in a vector should be of the same type.

Matrix - a vector with two-dimensional shape information Matrix is a special kind of vector. A matrix is a vector with two additional attributes: the number of rows and the number of columns.

Array: Similar to matrix, but arrays can have more than two dimensions.

**What’s the difference between tapply and aggregate?**

Rubric

1 points: On your presentation day, load your slides on the instructor's computer before class starts. Bring your slides on a thumbdrive (multiple people can share a thumb drive).

1 points: You name is clear and bold on the title and closing slide

3 points: your presentation is more than 7 minutes and less than 9 minutes (this gives a little time for questions)

3 Points: you must teach us something about visualization (a technique, a useful R package, a tool we haven't covered in class, or something about the history of info vis)

3 Points: You show us examples you have created yourself

2 points: presentation is practiced - not too fast, smooth delivery, no technical glitches

2 Points: slides are well designed without too much text

You can earn from 1 to 5 NEGATIVE POINTS on your own presentation for talking during others' presentations

You will lose 5 points automatically if you are not ready on the day you are assigned

POST YOUR SLIDES AS A PDF TO BLACKBOARD BEFORE CLASS or the assignment will be counted late.

IMPORTANT:

Demos are BAD. They often don't work on class machines. They take more time then expected. They are one more moving part that can go wrong in so many ways! Avoid demos when possible.

If you are showing us a web page, try it out on multiple machines, particularly on a lab machine. Generally avoid.

No videos. No videos. No videos.

A single slide takes about one minute to present. After the title slide, plan to have 7 or 8 slides of content and one "Thank You" slide (which takes zero minutes).

Practice your presentation at least 3 times, out loud. Better if you do in front of others or a mirror.

Avoid too much text on a slide. Better to use a few small images that remind you what to talk about.