## How can I extract elements from lists of lists in R?

Asked 5 years, 5 months ago Active 3 years, 4 months ago Viewed 19k times



I have a bunch of lists containing lists within them (generalised linear model output). I want to write a function which will extract several elements from each list and then combine the results into a data frame.



I want to extract modelset[[1]] slikelihood & modelset[[1]] fixef, modelset[[2]] slikelihood & modelset[[2]] fixef, etc, and combine the results into a data frame.



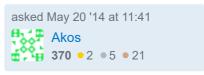
Can someone give me an idea of how to do this?

Apologies if my question is confusing: what I am trying to do is beyond my limited programming understanding.

Further information about my list:

r list





```
2 lapply( modelset , "[[" , "likelihood" ) etc... - Simon O'Hanlon May 20 '14 at 11:51
```

## 1 Answer



In order to solve this elegantly you need to understand that you can use ['...'] instead of \$... to access list elements (but you will get a list back instead of an individual element).

52

So if you want to get the elements likelihood and fixef, you can write:



```
modelset[[1]][c('likelihood', 'fixef')]
```

By using our site, you acknowledge that you have read and understand our <u>Cookie Policy</u>, <u>Privacy Policy</u>, and our <u>Terms of Service</u>.

```
lapply(modelset, function (x) x[c('likelihood', 'fixef')])
```

This works, but it's not very R-like.

You see, in R, almost **everything** is a function. [...] is calling a function named [ (but since [ is a special symbol for R, in needs to be quoted in backticks: `[`). So you can instead write this:

```
lapply(modelset, function (x) `[`(c('likelihood', 'fixef')])
```

Wow, that's not very readable at all. However, we can now remove the wrapping anonymous function (x), since inside we're just calling another function, and move the extra arguments to the last parameter of lapply:

```
lapply(modelset, `[`, c('likelihood', 'fixef'))
```

This works and is elegant R code.

Let's step back and re-examine what we did here. In effect, we had an expression which looked like this:

```
lapply(some_list, function (x) f(x, y))
```

And this call can instead be written as

```
lapply(some_list, f, y)
```

We did exactly that, with somelist = modelset, f = [] and y = c('likelihood', 'fixef').

answered May 20 '14 at 12:00



5 It not easy to find such didatic answer like this one. Thank you. – Paulo E. Cardoso May 20 '14 at 12:08

Thank you for your very helpful answer and explanation. - Akos May 20 '14 at 12:14

By using our site, you acknowledge that you have read and understand our <u>Cookie Policy</u>, <u>Privacy Policy</u>, and our <u>Terms of Service</u>.