- factor expansion – needs example of intended behavior

Let's look at `vapply()`. It's used as such (run this code):

```
vapply(mtcars, class, character(1))
vapply(list(matrix(1:9, nrow=3), matrix(1:20, nrow=5)), dim, numeric(2))
```

Exercise. Let `trims = seq(0, 0.5, 0.1)` and `x = rnorm(100)`. Rewrite the expression `lapply(trims, function(trim) mean(x, trim=trim))` to not need an anonymous function.[1]

In general, when calling `vapply(args, func, example)`, *each time `func()` is called on an element of `args`*, the output must have the same type and length as `example`. Otherwise, `vapply()` stops with an error. Also, when returning multiple numeric vectors, `vapply()` will add appropriate dimensions to the output.

Exercise. With a variety of different functions, test the behavior of `vapply()` and `sapply()` when the list of arguments is an empty list (`list()`). How would the behavior of `vapply()` help you write code robust to errors and bugs?[^bugs]

Exercise. What happens when `sapply(args, func)` is called in a situation where `func()` returns vectors of different lengths for different elements of `args`? How can `vapply()` be used to detect unexpected instances of this situation?

Exercise. Experiment with lists containing `Sys.time()`. In particular, what happens when you use an `*apply()` function to determine the class of every element in a list containing `Sys.time()` for one of its entries?

==========

Regional-level analysis

We'll also sometimes want to take a step back and group some of our observations together to do data analysis at a different level.

- Aggregate at the level of regions using the `aggregate()` function. (*Hint:* Pass in `FUN=median`.)
- Compute the correlations between the resulting columns.
- How do these compare with the correlations you calculated at the state level? What do you think explains the difference?

[1] Try `lapply(trims, mean, x=x)`.

Adding interaction terms

We can add *interaction terms* to a linear regression very easily: in the list of predictors we pass in to the `lm()` function, we can include the interaction of `var1` with `var2` by including `var1:var2` or `var1*var2`.

- What's the difference between including `var1:var2` or `var1*var2`? (*Hint:* Try regressing against nothing aside from the interaction term.)
- How much additional predictive power can you get by including well-chosen interaction terms in your regression? Which interaction terms help the most?