Lists

What is a list?

We have seen lists come up a few times in class so far.

- We can uses lists to return multiple arguments from a function
- A data frame is technically a list
- The output of some commands to read in `HFD5` files

A list is a special type of vector that can contain arbitrary objects as elements.

What's special about working with lists?

Elements of a list are accessed with:

- A dollar sign $ followed by the name of the element
- Double brackets followed by the index of the corresponding element of the list, e.g. `[[1]]` to access the first element

Note: We can also subset a list, but for this we will use single brackets and treat the list as a vector. For instance, `[1:8]` would subset a list with more than 8 elements to a list of just the first 8 elements.

How do we work with lists?

- `lapply` applies the same function to every element of a list
- `do.call` applies a function using every element of a list as arguments
- `unlist` takes a list and forces it to become a vector

HFD5 Example

```
library(rhdf5)
nasa <- H5Fopen("~/Downloads/nasa.h5")
data <- nasa$'tile10'
```

The object `data` is a list. How many elements does it have, and what types of objects are they?

HFD5 Example

There are two ways to access the first element of the list

- `data$01_LAI`
- `data[[1]]`

We've seen this before when we were learning about reading in this data.

HFD5 Example

What if we wanted to summarize the mean of the first 18 elements of the list?

```
means <- lapply(data[1:18], mean)
```

What type of object is `means`?

What happens if we apply `unlist(means)`?

HFD5 Example

What if we wanted to treat each 1000×1000 matrix of measurements as a variable and construct a $1000^2 \times 19$ data frame?

```
df <- do.call("cbind", lapply(data, "c"))
```

HFD5 Example

```
h5closeAll()
```

How to Make a List

You can create a named list by using the `list` function and supplying a sequence of objects (optionally) preceded by names and an equals sign, e.g.

```
mylist <- list("ns" = 0:9,
               "ls" = c("a", "b", "c"))
```