Functionals

Functionals

- A **functional** is a function that takes a function as input and returns a vector.
- Functionals are used to abstract over common patterns of looping.
- Common functions are `lapply()`, `apply()`, `tapply()`, ...
- Reduce bugs by better communicating intent.

```
set.seed(1014)

# Create some random output:
# 20 random vectors with random lengths
l <- replicate(20, runif(sample(1:10, 1)),
  simplify = FALSE)
str(l)
l
```

```r
# Extract length of each element
lengths <- vector("list", length(l))
for (i in seq_along(l)) {
  lengths[[i]] <- length(l[[i]])
}
lengths
```

Preallocating space for output saves a lot of time

```
# Extract length of each element
lengths <- vector("list", length(l))
for (i in seq_along(l)) {
  lengths[[i]]
}
lengths
```

Safe shortcut for `1:length(l)`

How would you change this to compute the mean of each element?

```
compute_length <- function(x) {
  out <- vector("list", length(l))
  for (i in seq_along(l)) {
    out[[i]] <- length(l[[i]])
  }
  out
}
```

How would you change this to compute the median of each element?

```
compute_mean <- function(x) {
  out <- vector("list", length(l))
  for (i in seq_along(l)) {
    out[[i]] <- mean(l[[i]])
  }
  out
}
```

How would you change this to compute the median of each element?

```
compute_median <- function(x) {
  out <- vector("list", length(l))
  for (i in seq_along(l)) {
    out[[i]] <- median(l[[i]])
  }
  out
}
```

```
# How would you reduce the duplication here?
f1 <- function(x) x + 1
f2 <- function(x) x + 2
f3 <- function(x) x + 3
```

Functions can be arguments!

```
compute <- function(x, f) {
  out <- vector("list", length(x))
  for(i in seq_along(x)) {
    out[[i]] <- f(x[[i]])
  }
  out
}

compute(l, length)
compute(l, mean)
compute(l, median)
```

Placeholder for “any other” arguments

```
compute <- function(x, f, ...) {
  out <- vector("list", length(x))
  for(i in seq_along(x)) {
    out[[i]] <- f(x[[i]], ...)
  }
  out
}

compute(l, mean, trim = 0.5)
compute(l, mean, na.rm = TRUE)
```