Lecture 8: Lists

Iterating over functions

So far:

```r
set.seed(45)

# Simulate from a N(0,1)
assess_coverage(n = 100, nsim = 1000, beta0 = 0.5, beta1 = 1,
                noise_dist = rnorm)
```

```
[1] 0.949
```

```r
# Simulate from Exp(1)
assess_coverage(n = 100, nsim = 1000, beta0 = 0.5, beta1 = 1,
                noise_dist = rexp)
```

```
[1] 0.96
```

What if I want to simulate from *many* distributions?

Idea

Vectors revisited

Vectors can contain numbers, booleans, characters, etc:

```r
x <- c(0, 1, 2)
x
```

```
[1] 0 1 2
```

```r
typeof(x)
```

```
[1] "double"
```

```r
x <- c("a", "b", "c")
x
```

```
[1] "a" "b" "c"
```

```r
typeof(x)
```

```
[1] "character"
```

The `typeof` function tells what *type* of object we have

Vectors of multiple types?

```r
x <- c(0, 1, "a")
x
```

```
[1] "0" "1" "a"
```

```r
x[1] + 1
```

```
Error in x[1] + 1: non-numeric argument to binary operator
```

Basic vectors (called *atomic* vectors) only contain one type.

Lists

```r
x <- list(c(0, 1), "a")
x
```

```
[[1]]
[1] 0 1

[[2]]
[1] "a"
```

Lists

```r
x <- list(c(0, 1), "a")
x
```

```
[[1]]
[1] 0 1

[[2]]
[1] "a"
```

```r
x[[1]]
```

```
[1] 0 1
```

```r
x[[1]][1]
```

```
[1] 0
```

Lists

```r
x <- list(c(0, 1), "a")
x
```

```
[[1]]
[1] 0 1

[[2]]
[1] "a"
```

```r
x[[1]]
```

```
[1] 0 1
```

```r
x[[1]][1]
```

```
[1] 0
```

```r
typeof(x[[1]])
```

```
[1] "double"
```

```r
x[[2]]
```

```
[1] "a"
```

```r
typeof(x[[2]])
```

```
[1] "character"
```

Visualizing list structure

```r
x1 <- list(c(1, 2), c(3, 4))
x1
```

```
[[1]]
[1] 1 2

[[2]]
[1] 3 4
```

```r
x2 <- list(list(1, 2), list(3, 4))
x2
```

```
[[1]]
[[1]][[1]]
[1] 1

[[1]][[2]]
[1] 2



[[2]]
[[2]][[1]]
[1] 3

[[2]][[2]]
[1] 4
```

Indexing lists

```r
x <- list(c(1, 2), c(3, 4))

x[1]
```

```
[[1]]
[1] 1 2
```

```r
typeof(x[1])
```

```
[1] "list"
```

```r
x[[1]]
```

```
[1] 1 2
```

```r
typeof(x[[1]])
```

```
[1] "double"
```

- `x[1]` returns a *list* which contains the first component of `x`
- `x[[1]]` returns the object stored in the first component

Indexing lists

```r
x <- list(list(1, 2), list(3, 4))
x[1]
```

Question: What will `x[1]` return?

Indexing lists

```r
x <- list(list(1, 2), list(3, 4))
x[1]
```

```
[[1]]
[[1]][[1]]
[1] 1

[[1]][[2]]
[1] 2
```

Indexing lists

```r
x <- list(list(1, 2), list(3, 4))
x[[1]]
```

Question: What will `x[[1]]` return?

Indexing lists

```r
x <- list(list(1, 2), list(3, 4))
x[[1]]
```

```
[[1]]
[1] 1

[[2]]
[1] 2
```

Question: How do I get just the 3?

Indexing lists

```r
x <- list(list(1, 2), list(3, 4))
x[[2]][[1]]
```

```
[1] 3
```

Vectors of functions?

Can we make a vector of *functions*?

```r
chisq_1 <- function(m){
  return(rchisq(m, df=1))
}

x <- c(rexp, rnorm, chisq_1)
x
```

```
[[1]]
function (n, rate = 1)
.Call(C_rexp, n, 1/rate)
<bytecode: 0x7f7c1c2cee08>
<environment: namespace:stats>

[[2]]
function (n, mean = 0, sd = 1)
.Call(C_rnorm, n, mean, sd)
<bytecode: 0x7f7c1a7c9ba8>
<environment: namespace:stats>

[[3]]
function(m){
```

Lists of functions

```r
x <- list(rexp, rnorm, chisq_1)
x[1]
```

```
[[1]]
function (n, rate = 1)
.Call(C_rexp, n, 1/rate)
<bytecode: 0x7f7c1c2cee08>
<environment: namespace:stats>
```

```r
x[1](10)
```

```
Error in eval(expr, envir, enclos): attempt to apply non-function
```

Question: Why does this cause an error?

Lists of functions

```r
x <- list(rexp, rnorm, chisq_1)
x[[1]]
```

```
function (n, rate = 1)
.Call(C_rexp, n, 1/rate)
<bytecode: 0x7f7c1c2cee08>
<environment: namespace:stats>
```

```r
x[[1]](10)
```

```
 [1] 0.57913414 1.02951803 0.54312869 0.59578710 0.69527103 0.32545401
 [7] 0.04481333 3.96257222 1.35634369 0.87948643
```

Class activity

https://sta279-s24.github.io/class_activities/ca_lecture_8.html