Introducing vectors

Every variable is a vector. Here, e is a vector with N values. X1 is the sequence of integers from 1 through 10. Vectors can grow and shrink as needed.

```
N = 100
set.seed(1234)
e = rnorm(N, mean = 0, sd = 1)
X1 = 1:10
```

Vector construction

Vectors are constructed in one of several ways

- Construction from a sequence
- Concatenation
- Return from a function or operation
- Growth by assignment

Vector sequence

```r
i = 1:5
year = 2000:2010
pies = seq(from = 0, by = pi, length.out = 100)
```

Concatenation

```
i = c(1, 2, 3, 4, 5)
j = c(6, 7, 8, 9, 10)
k = c(i, j)
l = c(1:5, 6:10)
```

Return and growth

```
i = rep(pi, 100)
i = sample(1:10, 4)
# Growth by assignment
i[30] = 3
```

NA

In the previous step, what are the values in i[5:29]?

```
print(i[5])

## [1] NA

is.na(i[5])

## [1] TRUE

print(i)

##  [1]  7  5  3  6 NA NA NA NA NA NA NA NA NA NA NA NA NA
## [24] NA NA NA NA NA NA  3
```

Vector access - by index

Vectors may be accessed by their numeric indices. Remember, ':' is shorthand to generate a sequence.

```
e[1]

## [1] -1.207

e[1:4]

## [1] -1.2071  0.2774  1.0844 -2.3457

e[c(1, 3)]

## [1] -1.207  1.084
```

Vector access - logical access

Vectors may be accessed logically.

```
i = 5:8
i[c(TRUE, FALSE, FALSE, TRUE)]

## [1] 5 8

i[i > 6]

## [1] 7 8
```

which

The which function return the indices where a logical expression is TRUE.

```
i = 5:8
which(i > 6)

## [1] 3 4

i[which(i > 6)]

## [1] 7 8

b = i > 6
i[b]

## [1] 7 8
```

Vectors may be used in arithmetic operations. Y is now a vector with length equal to the longest vector used in the calculation.

```
B0 = 5
B1 = 2.25
X1 = rep(seq(1, 10), 10)
Y = B0 + B1 * X1 + e
```

Question: B0 and B1 are vectors of length 1. X1 and e are vectors of length 100. How are they combined?

Recycling

R will "recycle" vectors until there are enough rows to perform an operation, i.e. everything gets as "long" as the longest vector in the operation. For scalar operations on a vector this doesn't present any issues.

Recycling - 2

Try the following code:

```
vector1 = 1:10
vector2 = 1:5
scalar = 3

print(vector1 + scalar)

##  [1]  4  5  6  7  8  9 10 11 12 13

print(vector2 + scalar)

## [1] 4 5 6 7 8

print(vector1 + vector2)

##  [1]  2  4  6  8 10  7  9 11 13 15
```

Exercises

- Create a vector of integers from 1 to 1000.
- Use logical access to generate a new sequence of the even numbers from 1 to 1000. (Hint: the modulus operator in R is % %.)
- Create a vector of e raised to the power of 1:20.
- Create a vector of length 300 with a random sample of the integers from 1 to 1000. Sample with replacement.

Solution

```
i = 1:1000
j = i[i%%2 == 0]
k = exp(1:20)
l = sample(1:1000, 300, replace = TRUE)
```