Econometrics 1 *Applied Econometrics with R*

Lecture 5: Programming with R (2)

黄嘉平

中国经济特区研究中心 讲师

办公室：文科楼2613

E-mail: huangjp@szu.edu.cn

Tel: (0755) 2695 0548

Website: https://huangjp.com

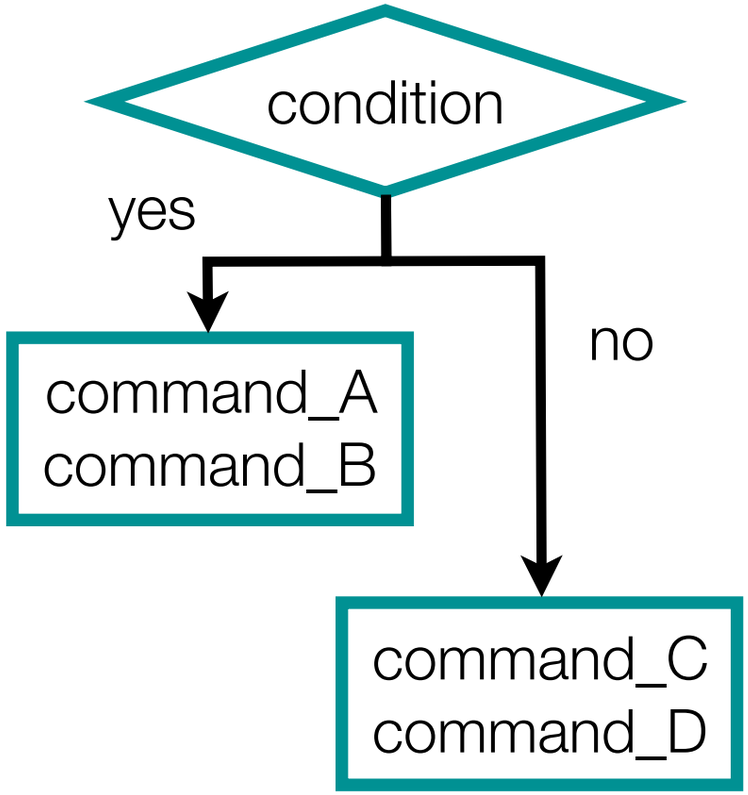
Review

- Function

```
function_name <- function (x) {
   ……
   ……
   return(…)
}
```

- Condition: the `if` statement

```
if (condition) {
   command_A
   command_B
   ……
} else {
   command_C
   command_D
   ……
}
```



Practice: the real cube root

- Define a function named `rcbrt` which always returns the real cube root of a real number (either positive or negative).

- For example, you may expect `rcbrt(8)` returns 2, and `rcbrt(-8)` returns -2.

- Use `if` statement in your function.

Practice: the real cube root

```
rcbrt <- function (x) {
   if (x < 0) {
      r <- -abs(x)^(1/3)
   } else {
      r <- x^(1/3)
   }
   return(r)
}
```

Iteration (loop): the `for` statement

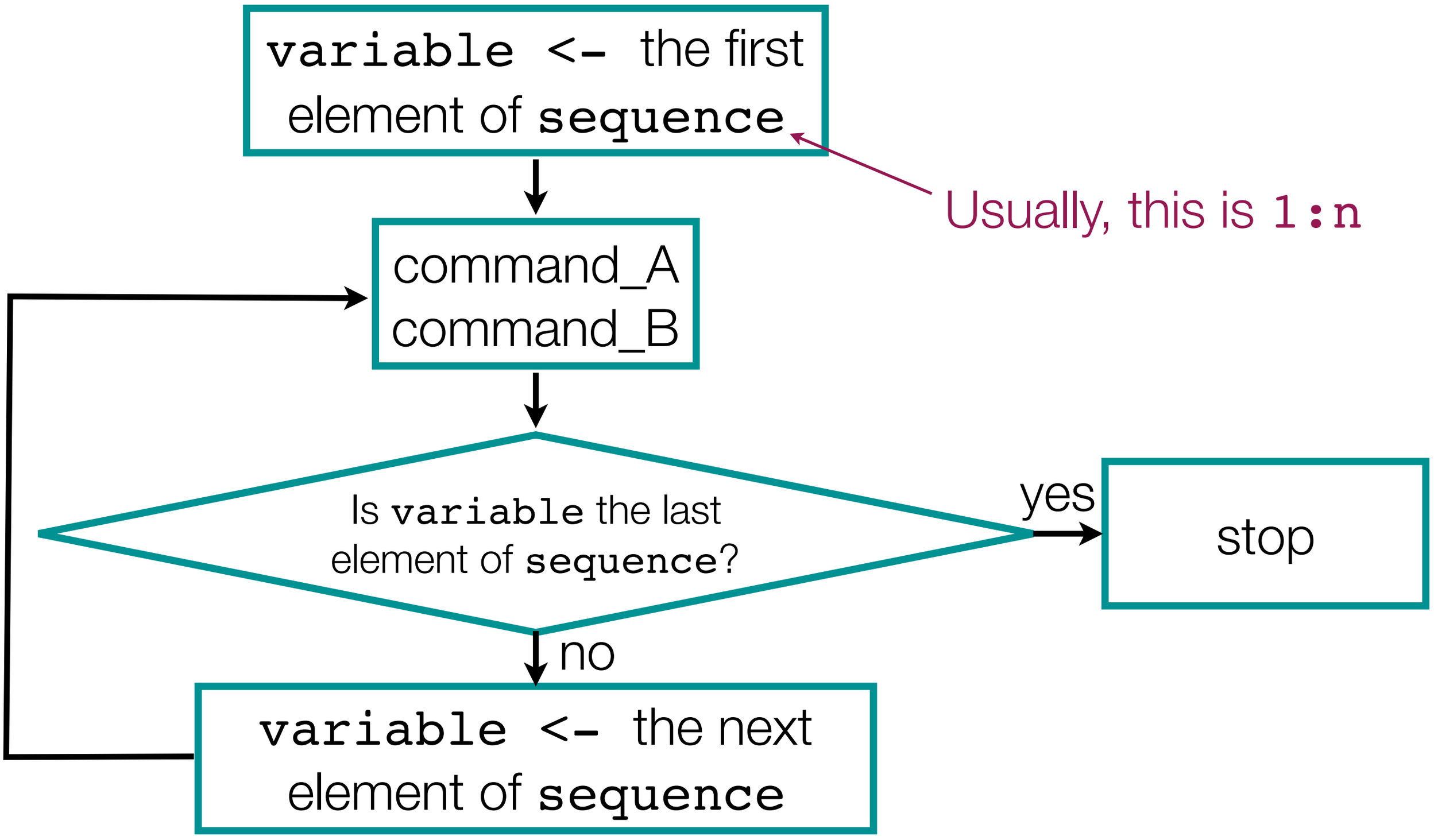
- If you want to repeatedly do something, or apply the same commands to different objects, you can use the `for` statement.

```
> for (variable in sequence) commands↩
```

Multiple lines

```
for (variable in sequence) {
    command_A
    command_B
    ……
}
```

Iteration (loop): the `for` statement



Apply a calculation to every element of a vector

```
> x <- 1:10↩

> y <- rep(0, 10)↩

> for (i in 1:length(x)) y[i] <- 1/x[i]↩
```

```
> z <- 0↩

> for (i in 1:length(x)) {↩
      z <- z + x[i] * y[i]↩
  }↩
```

Example: calculate the factorial $f(n) = n!$

```
fac <- function (n) {
  if (n == 0) return(1)

  f <- 1
  for (i in 1:n) {
    f <- f * i
  }
  return(f)
}
```

Practice: the heights of students

1. Collect data of the heights of students in the class.
2. Put the data into a vector.
3. Count the number of students that are taller than 160cm.
   (Use `for` statement in your program.)

- Note: an easy solution is `sum(x > 160)`

Practice: the heights of students

```
x <- ……

ntall <- 0

for (i in 1:length(x)) {
  if (x[i] > 160) ntall <- ntall + 1
}
```

Practice: matrix multiplication

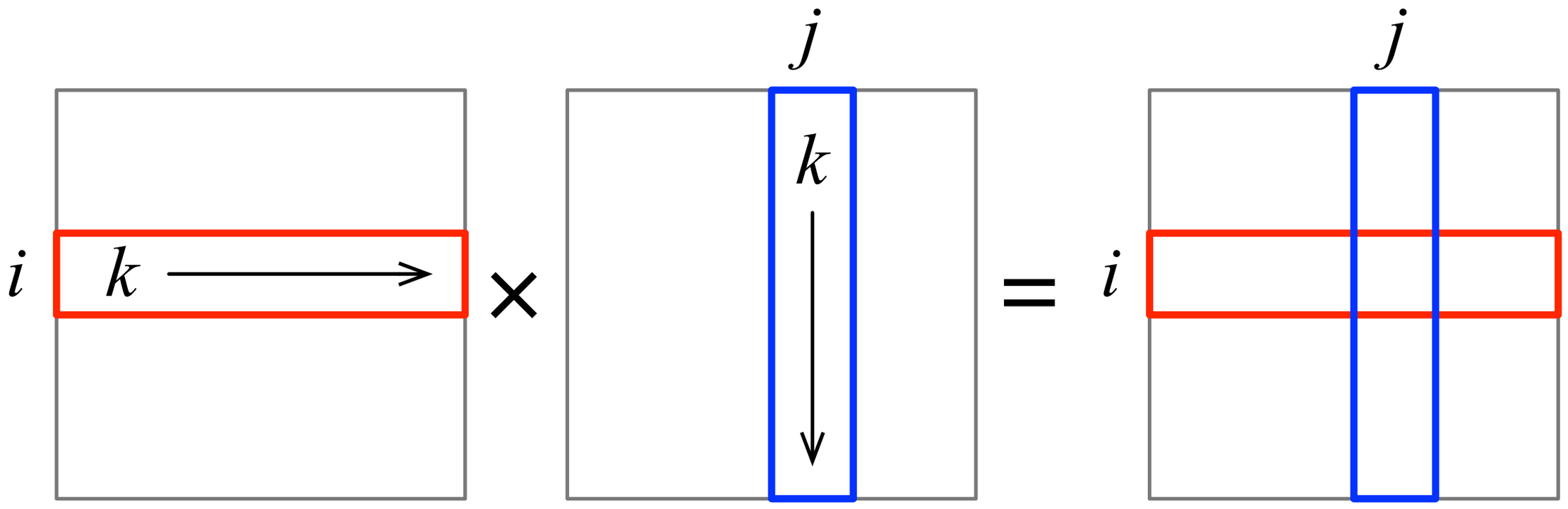
- Write a function that calculates the product of two square matrices with the same size using `for` statement (and without using `%*%` command). Create two matrices of arbitrary size and test your function.

- For example:

```
matmul <- function (a, b) {
      ……
      ……
      return(…)
}
```

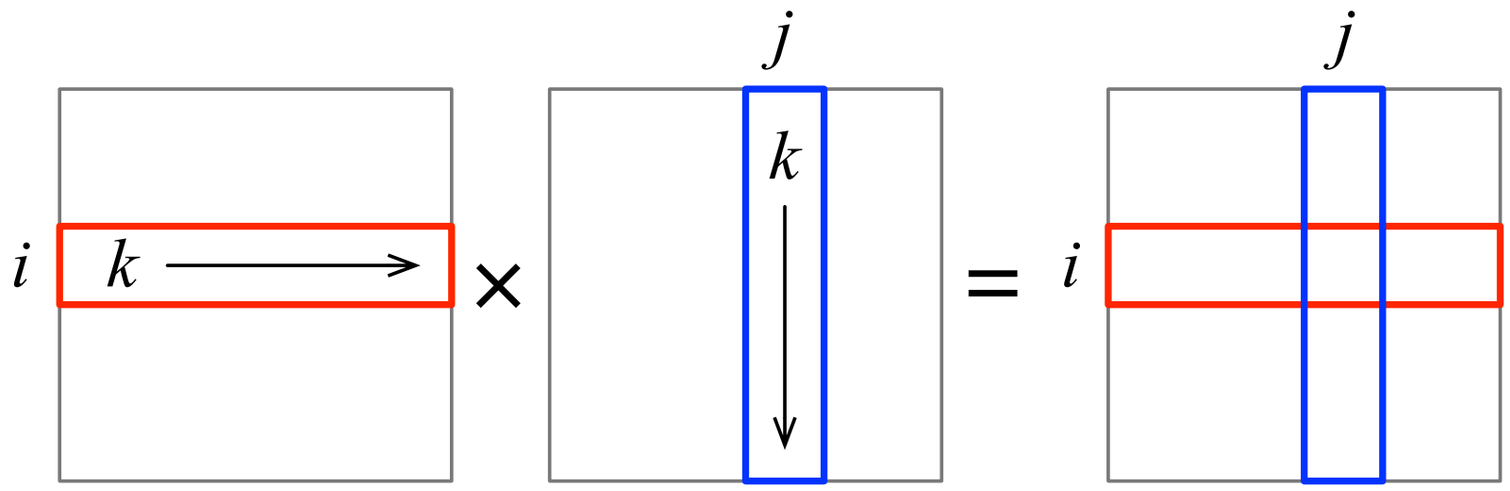
Practice: matrix multiplication

$$A \times B = C \quad \Leftrightarrow \quad C_{ij} = \sum_k A_{ik} \times B_{kj}$$



Practice: matrix multiplication

```
matmul <- function (x, y) {
   n <- nrow(x)
   z <- matrix(rep(0, n^2), n, n)
   for (i in 1:n) {
      for (j in 1:n) {
         for (k in 1:n) {
            z[i,j] <- z[i,j] + x[i,k] * y[k,j]
         }
      }
   }
   return(z)
}
```



Practice: random walk

- A stochastic process $Z(t)$ is called a random walk if

$$Z(0) = 0 \quad \text{and} \quad Z(t) = Z(t\text{-}1) + B \quad \text{for } t = 1, 2, \ldots,$$

  where B is a random variable that takes value 1 with probability p and -1 with probability $(1\text{-}p)$. Generate a random walk with 200 periods (e.g. $t = 0, \ldots, 200$) and $p = 0.6$, and plot your results.

- Use `rbinom()`