Chapter 6:
Loops and Conditions

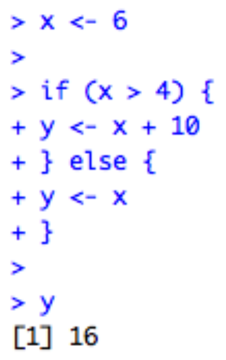
Conditional tests

✓ If…then…else

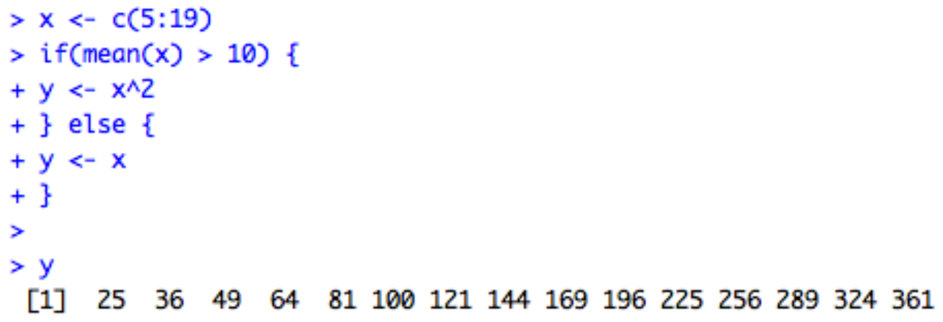
- ➡ This construct takes the form

  if (expression 1) {construct 1} else {construct 2}

- ➡ Examples:



```
> x <- 6
>
> if (x > 4) {
+ y <- x + 10
+ } else {
+ y <- x
+ }
>
> y
[1] 16
```



```
> x <- c(5:19)
> if(mean(x) > 10) {
+ y <- x^2
+ } else {
+ y <- x
+ }
>
> y
 [1]  25  36  49  64  81 100 121 144 169 196 225 256 289 324 361
```

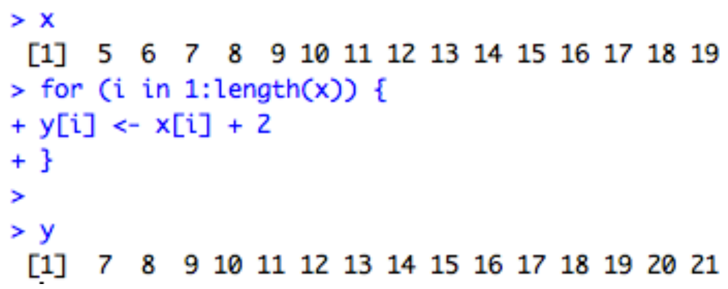
Repetition

✓ For

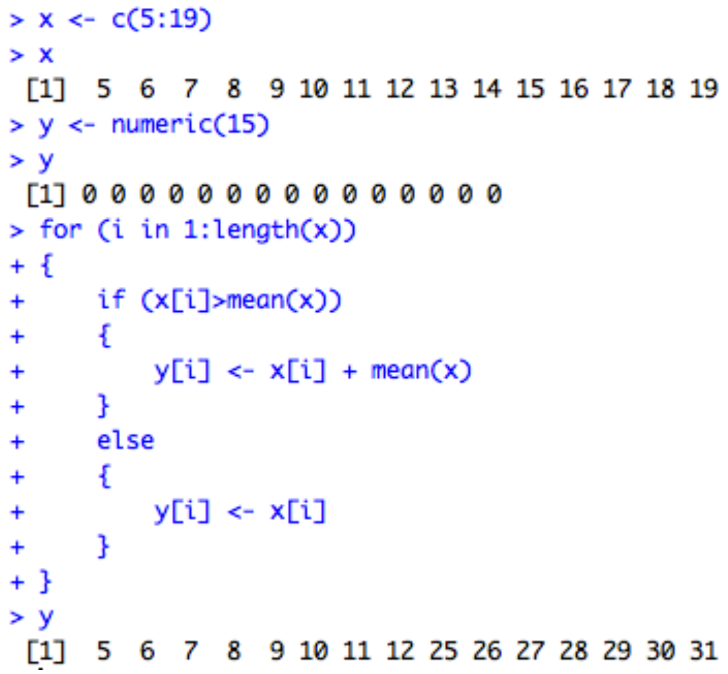
➡ This construct takes the form

for (name in expression 1) {construct 1}

➡ Examples:



```
> x
 [1]  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
> for (i in 1:length(x)) {
+ y[i] <- x[i] + 2
+ }
>
> y
 [1]  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
```



```
> x <- c(5:19)
> x
 [1]  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
> y <- numeric(15)
> y
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> for (i in 1:length(x))
+ {
+     if (x[i]>mean(x))
+     {
+         y[i] <- x[i] + mean(x)
+     }
+     else
+     {
+         y[i] <- x[i]
+     }
+ }
> y
 [1]  5  6  7  8  9 10 11 12 25 26 27 28 29 30 31
```

➡ R allows for nesting one loop or conditional statement within another as seen in example 2

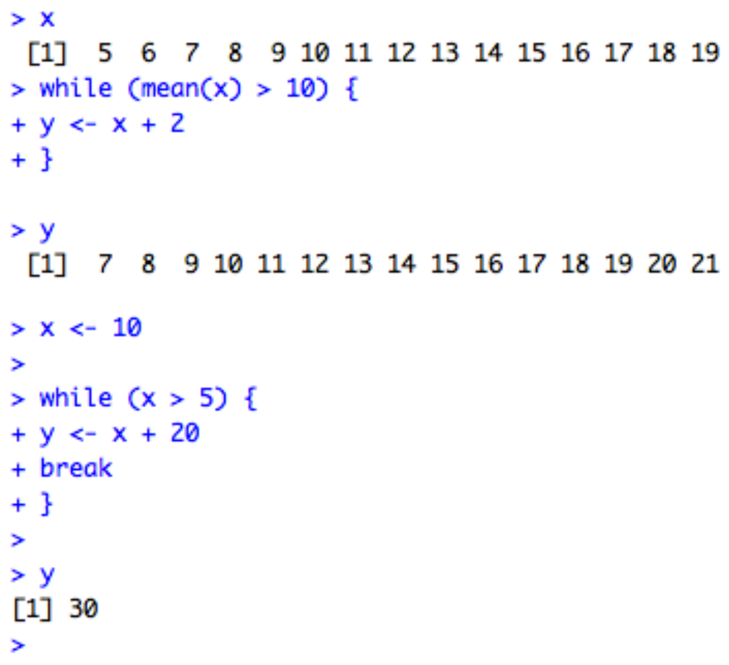
Repetition

✓ While

- ➡ This construct takes the form

  while (expression 1) {construct 1}

- ➡ Examples:



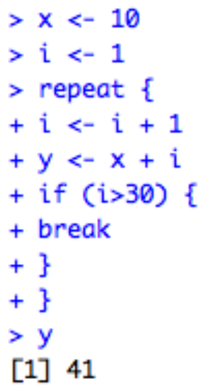
```
> x
 [1]  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
> while (mean(x) > 10) {
+ y <- x + 2
+ }

> y
 [1]  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21

> x <- 10
>
> while (x > 5) {
+ y <- x + 20
+ break
+ }
>
> y
[1] 30
>
```

- ➡ In the second example, because x will always be greater than 5, R stays in the loop, continuously assigning x + 20 to y. The *break* statement is used to stop this from occurring

Repetition

- ✓ Repeat
  - ➡ This construct takes the form

    repeat {construct 1}
  - ➡ Example:



```
> x <- 10
> i <- 1
> repeat {
+ i <- i + 1
+ y <- x + i
+ if (i>30) {
+ break
+ }
+ }
> y
[1] 41
```