R Programming: Week 2 - Controlling Flows

Objectives

Write an if-else expression Write a for loop, a while loop, and a repeat loop Define a function in R and specify its return value [see Functions Part 1 and Part 2] Describe how R binds a value to a symbol via the search list Define what lexical scoping is with respect to how the value of free variables are resolved in R Describe the difference between lexical scoping and dynamic scoping rules. Convert a character string representing a date/time into an R datetime object. [see Dates and Times]

If/Else

If/Else tests conditions.

Basics

Example 1:

```
x <- 4
if(x > 3) { y <- 10} else y <- 0;
y
```

```
## [1] 10
```

Example 2:

```
x=0
y <- if(x > 3) {10
} else {
 0
}
```

Example 1 and 2 produce the same output.

Can use just if/if statements

For Loops

For loops execute a loop a fixed number of times.

Example 1: Basic

```
for(i in 1:10) {
 print(i)
};
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

Example 2: All do the same thing

```
x <- c("a", "b", "c", "d")
for(i in 1:4) {
 print(x[i])
}

## [1] "a"
## [1] "b"
## [1] "c"
## [1] "d"

for(i in seq_along(x)) {
 print(x[i])
}

## [1] "a"
## [1] "b"
## [1] "c"
## [1] "d"

for(letter in x) {
 print(letter)
}

## [1] "a"
## [1] "b"
## [1] "c"
## [1] "d"

for(i in 1:4) print(x[i]);

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

While Loops

While Loops execute a loop while a condition is true

Example 1: Basic While

```
count <- 0
while(count < 10) {
 print(count)
 count <- count + 1
};
```

```
## [1] 0
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
```

Example 2: Multiple Conditions

```
z <- 5
while(z >= 3 && z <= 10) { print(z)
  coin <- rbinom(1, 1, 0.5)
 if(coin == 1) { ## random walk
 z <- z + 1
 } else {
 z <- z - 1
 }}
```

```
## [1] 5
## [1] 4
## [1] 3
## [1] 4
## [1] 3
```

Repeat, Next, and Break

-Repeat executes an infinite loop -Is not heavily used in R generally -Next skips an iteration of a loop -Break breaks the execution of a loop - Used to exit a function

Example 1: Next

```
for(i in 1:100) {
 if(i <= 20) {
 ## Skip the first 20 iterations
 next
 }
 ## Do something here
};
```

Helpful hints -Avoid infinite loops -If working in commane line, use *apply functions.