

## R Programming: Week 2 - Functions

### 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]

### Functions

Example 1:

```
add2 <- function(x , y ) {  
  x + y  
}  
add2(3, 5)  
## [1] 8
```

Example 2:

```
above10 <- function(x){  
  use <- x > 10  
  x[use]  
}  
  
above <- function(x,n) {  
  use <- x > n  
  x[use]  
}  
x <- 1:20  
above(x, 12)  
## [1] 13 14 15 16 17 18 19 20
```

Example 3:

```
columnmean <- function(y){  
  nc <- ncol(y)  
  means <- numeric(nc)  
  for(i in 1:nc) {  
    means[i] <- mean(y[,i], na.rm = removeNA)  
  }  
}
```

```
means  
}
```

and colmeans(airquality, FALSE)

## Basics

Functions are used to store R Objects -Considered to be first class objects -Include formal argument

Functions arguments can be partially matched. - Order of operations - Check for exact match for a name argument - Check for a partial match - Check for a positional match

Example 4: Defining a function

```
f <- function(a, b = 1, c = 2, d = NULL) {  
}  
f  
  
## function(a, b = 1, c = 2, d = NULL) {  
## }
```

Example 5: Lazy Evaluation (arguments not being properly defined)

```
f <- function(a ,b) {  
  a^2  
}  
f(2)  
  
## [1] 4
```

Example 6: No Good Lazy Evaluation ``{r} f <- function (a,b) { print(a) print(b) } f(45) ``

... Argument - Indicates a variable number of arguments and is often used to not copy entire argument list of the original function - Is necessary when the number of arguments passed to the function cannot be known in advance.

Example 7: ... Argument

```
myplot <- function(x,y,type = "l",...) {  
  plot(x,y,type=type,...)  
}
```