

Functions and Control Flow

Paul M. Magwene

Functions

- ▶ Functions are a programming tool for organizing and abstract related computations.
- ▶ They help to logically structure analyses and programs.

To write your own function in R, define a function like this:

```
func.name <- function(arg) {  
  # one or more statements  
  # return final result (if any) as last line  
}
```

Example functions

```
add.two <- function(x) {  
  return(x + 2)  
}
```

The function `add.two`:

- ▶ takes a single argument, labelled `x`
- ▶ returns the value `x + 2`

return() is optional

```
add.two <- function(x) {  
  x + 2  
}  
  
rev.seq <- function(x){  
  rev(seq(x))  
}
```

- ▶ The value of the last expression is automatically returned so simple functions don't need a return statement

Arguments are placeholders for input

```
add.two <- function(x) {  
  x + 2  
}  
  
x <- 5  
add.two(3)  # returns 6 not 10  
[1] 5
```

Functions can take multiple arguments

```
squared.difference <- function(x, y) {  
  (x - y)**2  
}
```

```
squared.difference(10, 5)  
[1] 25
```

Arguments can have default values

```
squared.difference <- function(x, y=0) {  
  (x - y)**2  
}
```

```
squared.difference(10)      # y defaults to zero
```

```
[1] 100
```

```
squared.difference(10, 5)   # y is 5, inferred by position
```

```
[1] 25
```

```
squared.difference(10, y=5) # y is explicitly set as 5
```

```
[1] 25
```