Compound Expressions

R Programming Structures

Gaston Sanchez

R Coding Compendium

Donate

Introduction

Before describing some of the common programming structures in R, we need to talk about a basic concept called **Expressions**.

You've been using simple expressions so far, but we need to introduce the notion of a compound expression.

R Expressions

Expressions

R code is composed of a series of **expressions**

- assignment statements
- arithmetic expressions
- function calls
- conditional statements
- etc

Simple Expressions

You've been writing several simple expressions like the following ones:

```
# assignment statement
a <- 12345

# arithmetic expression
525 + 34 - 280

# function call
median(1:10)
```

Grouping Expressions

Constructs for grouping together expressions

- semicolons: `;`
- curly braces: `{}`

Separating Expressions

Simple expressions separated with new lines:

```
a <- 10
b <- 20
d <- 30
```

Grouping Expressions

Grouping simple expressions with semicolons (within a single line of text):

```
a <- 10; b <- 20; d <- 30
```

Although this is a perfectly valid expression, we recommend avoiding semicolons, since they make code harder to review.

Grouping Expressions

Another way to group expressions is by wrapping them within braces:

```
{
  a <- 10
  b <- 20
  d <- 30
}
```

R will treat this as one "unit" or "block" of code

Note: this piece of code is a perfectly valid expression, but I'm just using it for illustration purposes (don't write code like this!)

Grouping Expressions

Multiple expressions in one line within braces:

```
{a <- 10; b <- 20; d <- 30}
```

Note: again, this piece of code is just for illustration purposes (don't write code like this!)

Expressions

So far:

```
# Expressions can be simple statements:
5 + 3
```

```
## [1] 8
```

```
# Expressions can also be compound:
{5 + 3; 4 * 2; 1 + 1}
```

```
## [1] 2
```

Compound Expressions

- Compound expressions consist of multiple simple expressions
- Compound expressions require braces
- Simple expressions in a compound expression can be separated by semicolons or newlines

Simple Expressions

We use braces { } to group the statements of an expression:

```
# simple expression
{5 + 3}
```

```
## [1] 8
```

For simple expressions there is really no need to use braces.

Expressions

Recall that:

- A program is a set of instructions
- Programs are made up of expressions
- R expressions can be simple or compound
- **Every expression in R has a value**

Every expression has a value

Expressions

The value of an expression is the last evaluated statement:

```
# value of an expression
{5 + 3; 4 * 2; 1 + 1}
```

```
## [1] 2
```

The result has the visibility of the last evaluation

Compound Expressions

The variables inside the braces can be used in later expressions

```
{
  a <- "hi"
  print(2 + 2)
  mean(1:10)
}
```

```
## [1] 4
```

```
## [1] 5.5
```

What happens when R executes this code?

Compound Expressions

What about this code:

```
x <- {
  a <- "hi"
  print(2 + 2)
  mean(1:10)
}
```

```
## [1] 4
```

Compound Expressions

The variables inside the braces can be used in later expressions

```
x <- {
  a <- "hi"
  print(2 + 2)
  mean(1:10)
}
```

```
## [1] 4
```

```
a
```

```
## [1] "hi"
```

Compound Expressions

```
# simple expressions in newlines
z <- {
  x <- 4
  y <- x^2
  x + y}
x
```

```
## [1] 4
```

```
y
```

```
## [1] 16
```

```
z
```

```
## [1] 20
```

Repeat this Mantra

Every expression in R has a value: **the value of the last statement that was evaluated**

Every expression in R has a value: **the value of the last statement that was evaluated**

Every expression in R has a value: **the value of the last statement that wasevaluated**

Using Compound Expressions

So when do you use (compound) expressions?

We use compound expressions (i.e. single expressions wrapped within braces) in programming structures like:

- functions
- if-else conditionals
- iterations (loops)

Parenthesis, Brackets, and Braces

`()`	functions	`mean(1:10)`
`[]`	objects	`vec[3]` `mat[2,4]`

`{}`	compound expressions	`{`
		`  a <- 3`
		`  b <- a^2`
		`}`

Compound Expressions

Do not confuse a function call (having arguments in multiple lines) with a compound expression

```
# this is NOT a compound expression
plot(x = runif(10),
     y = rnorm(10),
     pch = 19,
     col = "#89F39A",
     cex = 2,
     main = "some plot",
     xlab = 'x',
     ylab = 'y')
```

Donation

If you find any value and usefulness in this set of slides, please consider making a one-time donation in any amount (via paypal). Your support really matters.

Donate