

# A Gentle Introduction to R

## EXTRAS

Jonathan Whiteley

2023-07-26

# Matrix math

- R can do *matrix math* — which are uses in many statistical procedures
  - ▶ But the *syntax* is different from the usual math operators
- Using a regular multiplication symbol (\*) results in *element*-wise multiplication
  - ▶ each *element* (item) in matrix1 is multiplied by the corresponding *element* in matrix2, etc.

```
c(1, 2, 3) * c(3, 2, 1)
```

- *Matrix multiplication* is specified by this operator: %\*%

```
c(1, 2, 3) %*% c(3, 2, 1)
```

# Symbolic Variables

- You can store values (*objects*) in symbolic variables (*names*) using an *assignment operator*

---

->	assign the <i>value</i> on the <b>left</b> to the <i>name</i> on the <b>right</b>
<-	assign the <i>value</i> on the <b>right</b> to the <i>name</i> on the <b>left</b>
=	assign the <i>value</i> on the <b>right</b> to the <i>name</i> on the <b>left</b>

---

- '<-' is preferred, because it is unambiguous (to people *and* to R)
- '=' is not allowed in certain situations (e.g., when surrounded by other expressions)
  - ▶ '=' is also used to set *argument values* in *function calls*, which is a different meaning and its most common use.
- You can also use the *assign function* (advanced):

```
assign('x', 3)      # assign the value 3 to the variable 'x'
```

# Variable / Object Names

- In R, all variables are *objects*
  - ▶ In R, **everything** is an *object*
- Object names can include:  
(depending on the language or *locale*)

---

letters	a-z A-Z
numbers	0-9
periods	.
underscores	_

---

- Names *should begin with a letter*

```
A <- 10
B = 10 * 10
log(A) -> A_log
B.seq <- 1:B
assign('x', 3)
```

# Object Names: Details

Names can start with a **letter** or a **period** (*more on this later*)

```
myvar <- T  
.myvar <- T
```

but anything else triggers an **error**

```
0myvar <- F  
_myvar <- F  
my var <- F
```

For more information about object names in R, see:

- Section 1.8 of 'An Introduction to R'
- Section 2.1.3 of 'The R Language Definition'

# Object Names: Hidden

- Names starting with a period (.) are special and normally hidden from users.

```
ls()  
ls(all.names = TRUE)
```

- Names starting with a period are used by packages or the system for special objects that users should not interact with directly.
- Such objects may not behave as expected with common commands, such as `ls()` (above).
- Therefore, most users should avoid doing this unless they know what they are doing and have a good reason to do so.

# Object Names: Advanced

- ‘Valid’ names following the rules above can be referred to easily in code.
- Names with any character are actually possible, but must be quoted with backticks (“`”)
  - ▶ **This is not recommended practice**, but occasionally useful when you need to refer to an element of an object, such as lists items or data frame columns, that have non-standard names.

```
`(my) [strange] {variable} 'name' "!@# $"` <- T  
print(`(my) [strange] {variable} 'name' "!@# $"`)
```

```
## [1] TRUE
```