A Gentle Introduction to R

EXTRAS

Jonathan Whiteley

2023-07-25

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)
```

Variable Names: Details

- Variable names containing only the following are easiest to work with:

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

- Names can start with a **letter** or a **period** *(more on this later)*, but anything else triggers an error

```
myvar = T
.myvar = T
```

These will fail:

```
0myvar = F
_myvar = F
my var = F
```

Variable 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 and data frames, that have non-standard names.

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

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

Variable Names: Hidden

- Variable names starting with a period (`.`) are special and normally hidden from users.

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

- Such variables are used by packages or the system for special values that users should not interact with directly.
- Such variables 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.