introduction to R

Vectors, Arithmetic, and Recycling

Learning Outcomes:

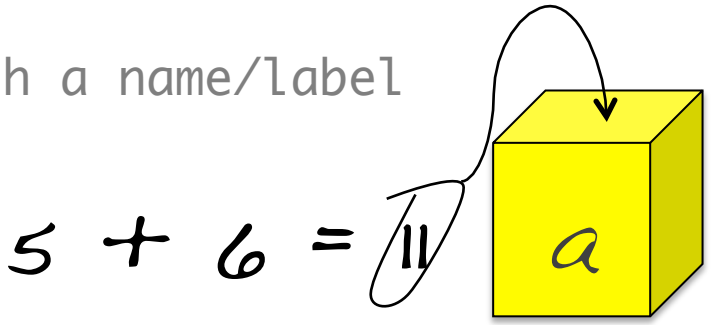
- use R as a calculator
- assign vectors to ***variables***
- be aware of different ***data types***
- get parts of a vector
- understand the ***recycling rule***

R as a calculator

```
# Arithmetic Operators
# + add, - subtract, * multiply, / divide
>  5 + 6
>  7 - 4
>  9 * 4
>  9 / 3

# Use parentheses to be explicit
>  4 + 3 * 2
>  (4 + 3) * 2

# power ^, modulo %%, integer division %/%
>  ??"arithmetic operators"
>  5 ^ 2
>  16 ^ (1/2)
>  7 %% 3
```

← evaluate and assign

```
# A variable is a 'box' with a name/label
>   a <- 5 + 6
```



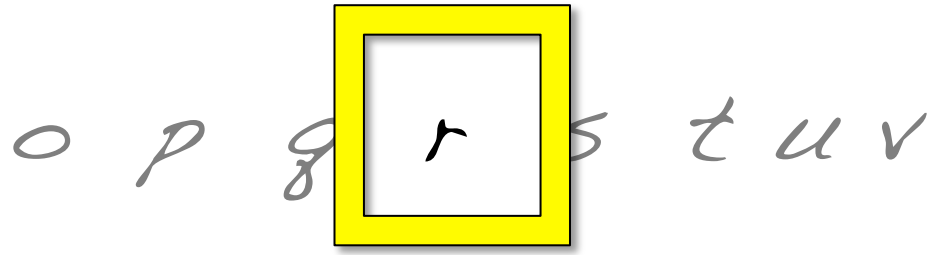

```
# Use the current value of the variable by typing its name
>   a + 9
>   11 + 9

# R is case sensitive so "a" is not the same as "A"
>   Apple <- a + 9
>   apple

# Assign one variable to another, and assign a new value
>   b <- Apple
>   Apple <- a^2
>   b
```

Data types

```
# Use typeof() to see what data type you have

"integer"        e.g.  1:10,  seq(1,10)

"double"         e.g.  0.25,  rnorm(5)

"char"        e.g.  "foo",  c("a","b","bar")

"logical"        e.g.  TRUE,  c(TRUE,FALSE),  c(F,F,F,T)


#  For more examples,

> ?typeof
```

[ ] indexing

```
#  Use the "built-in" vector letters
> letters
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m"
[14] "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"

#  Use [ ] to index (or subset) a vector
> letters[18]
[1] "r"
```




```
#  Use a vector argument – order matters
> my_letters <- letters[18:14]
> my_letters
[1] "r" "q" "p" "o"
```

Recycling rule

```
# Or, put elements together with c() = concatenate
>  long_vector <- c(1,3,2,5,12,3,0)

# multiplying by a scalar = element-wise multiplication
>  long_vector * 2
[1]  2  6  4 10 24  6  0

# R "recycles" the shorter vector
>  short_vector <- c(1,2,3)
>  long_vector * short_vector
```