Notes 3/15/19

Functions and Conditionals (if, then, and else)

Conditionals

* About if()
  + Whatever you write in parentheses will be coerced by R into some logical value, true or false ideally
    - Want to write something that will be a single true or a single false
  + If the content of if() is not something R can coerce to a logical you will get an error
* What if x has missing values (for function())?
  + Should include a condition na.rm=FALSE/TRUE
    - If(na.rm==TRUE) {

# reassign missing values using logical subsetting

X <- x[!is.na(x)]

}

* + - * Dont actually need ==TRUE
      * Can just write: if(na.rm) {}
* What if object x is not a vector?
  + If(is.vector(x) & type(x)==”numeric”) {}
* Running two lines in one line:
  + A=1; 2+2; 3\*3 will return

[1] 4

[1] 9

* + {a=1;2+2;3\*3} will only return [1] 9
    - But you can call a and get [1] 1
    - R doesn’t ignore it, but it is not displayed in the console

Executing based on assignments in an R expression:

* D <- {

A=1

B=2+2

3\*3

}

Ls()

[1] “a” “b” “c”

A

[1] 1

B

[1] 4

D

[1] 9

-the output of an R expression will always be the output of the very last thing that was executed in the expression, even if assigned to something Ex:

* D <- { A=1

B=2+2

}

D

[1] 4

* This means that if you leave what you want as output from a function as the last thing run in the expression, that will automatically be what is returned