Logic and Functions
UC Davis Statistics Club

Created by:
Christopher Wong
Academic Director

Logic

- Mathematical and logical symbols
  - Less than: $<$
  - Greater than: $>$
  - Less than or equal to: $<=$
  - Greater than or equal to: $>=$
  - Equals: $==$
  - Addition, subtraction, multiplication, division: **+**, **-**, *****, **/**
  - Modulus: **%%**
  - And: **&** and **&&**
  - Or: | and ||
    - & and | performs elementwise comparisons, && and || is more appropriate for control-flow (e.g. loops)
  - Not: **!**

Logical statements and Loops

- Logical statements
  - if(statement){code} evaluates if the statement is TRUE or FALSE. If TRUE, then the code is run.
  - else{code} comes after an if(statement) and the code is run when the if(statement) returns FALSE.
  - else if(statement){code} adds additional if(statement) to the code if the first if(statment) returns FALSE.
  - ifelse(statement, TRUE, FALSE) is a compact way of writing an if and else statement.
  - while(statement){code} continues to run until the statement is FALSE

- Loops
  - for(i in 1:length(foo)) For loops are not recommended as they are slow in R. Instead, we use
  - sapply(1:length(foo), function()) for vectors and data frames and
  - lapply(1:length(foo), function()) for lists
  - aggregate(data, by = list(), FUN = function) applies the function to the data after splitting them into subsets based on factors specified in by = list().

Writing your own functions

- R gives you the ability to write your own functions with **function()**

```
# This is where you will combine your R knowledge
# (subsetting, logical statements, loops, basic functions)
example.func = function(variables){
  Code goes here.
  return (Results)
}
```

- Data sampler/subsetter
- Probability and simulation
- Confidence intervals and visualization
- Data destroyer/cleaner
- Outlier finder

Bonus Topics

- Contrast of simulation and theory
  - Limits and infinite sums
  - Fibonacci numbers