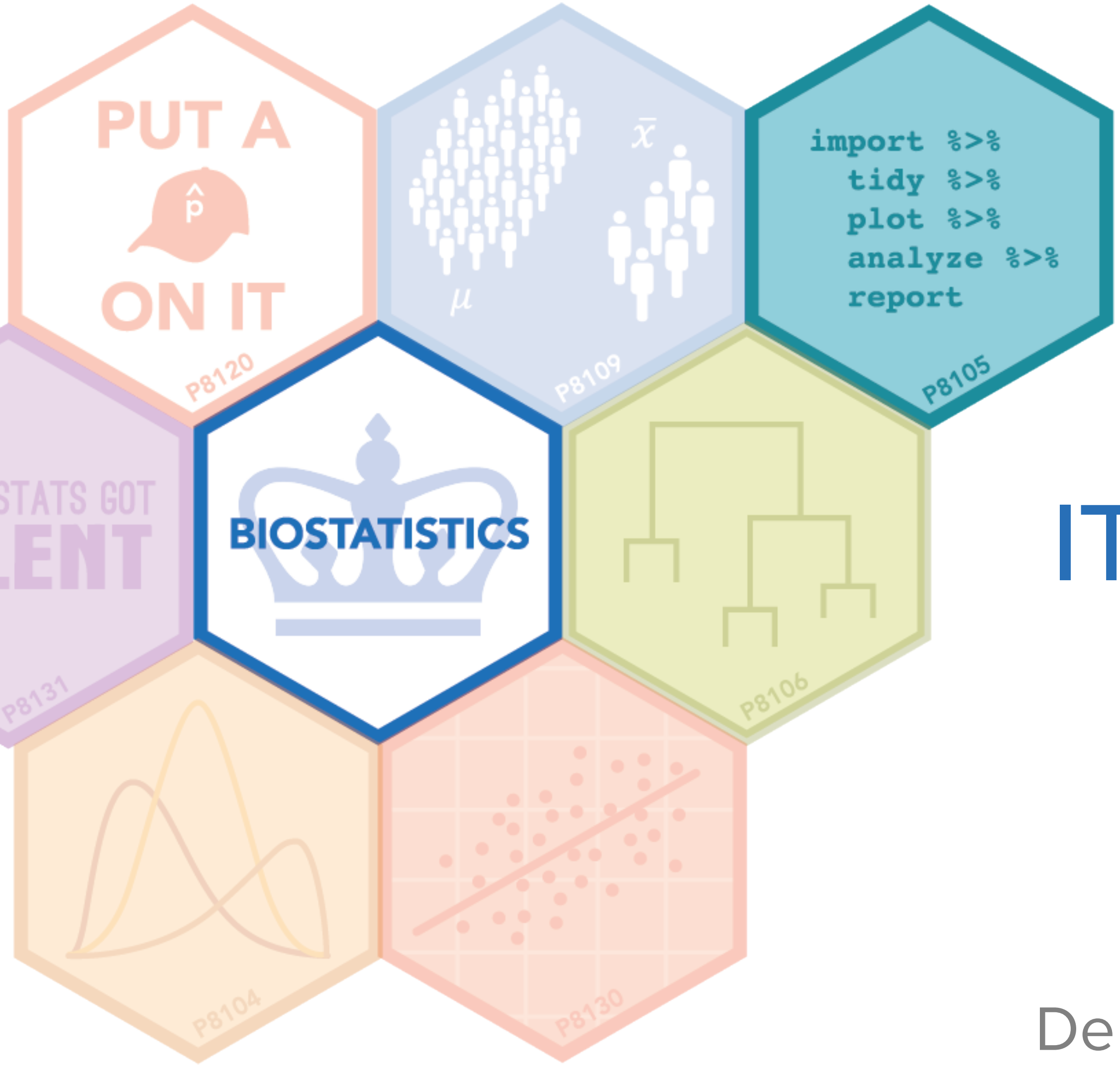

ITERATION AND SIMULATION

Jeff Goldsmith, PhD

Department of Biostatistics

Why iterate

- You will frequently encounter problems where you need to the same basic thing **a lot**
- The "don't write the same code more than twice" rule motivates the use of funtions
- The need to do the same thing a lot motivates formal structures for iterating

for loops

- Loops are the easiest place to start
- Loops consist of an output object; a sequence to iterate over; the loop body; and (optionally) an input object

- The basic structure is:

```
input = list(…)
output = list(…)

for (i in 1:n) {

    output[[i]] = f(input[[i]])

}
```

Loop functions

- The loop process (supply input vector / list; apply a function to each element; save the result to a vector / list) is really common

- For loops can get a little tedious, and a little opaque
  - Have to define output object and iteration sequence
  - Need to make sure loop body is indexed correctly
  - Often unclear on a first glance exactly how inputs are connected to outputs

- Loop functions are a popular way to clean up loops
  - We'll focus on `purrr::map()`
  - Base R has `lapply()` and similar functions

map

- Goal of map is to clarify the loop process

- The basic structure is

```
output = map(input, f)
```

- This produces the same result as the for loop, but emphasizes the input and function and reduces the amount of overhead
  - Doesn’t speed code up (as long as you have well-written loops)
  - Benefit comes from clarity

map variants

- By default, `map` takes one input and will return a list
- If you know what kind of output your function will produce, you can use a specific `map` variant to help prevent errors:
  - `map_dbl`
  - `map_lgl`
  - `map_df`
- If you need to iterative over two inputs, you can use `map` variants to give two input lists / vectors:
  - `map2`
  - `map2_dbl`
  - `map2_df`

Process

- I usually don’t jump straight to a function definition with a map statement to do iterative processes

- One workflow I use is
  - Write a single example for fixed inputs
  - Embed example in a for loop
  - Abstract loop body to a function
  - Re-write using a map statement

- This helps make each step clear, prevents mistakes, and only adds complexity when I need it

Repeated sampling

- “Repeated sampling” is a conceptual framework that underlies almost all of statistics
  - Repeatedly draw random samples of the same size from a population
  - For each sample, compute the mean
  - The distribution of the sample mean converges to a Normal distribution

- Repeated sampling doesn’t happen in reality
  - Data are difficult and expensive to collect
  - You get your data, and that’s pretty much it

- Repeated sampling can happen on a computer

Simulation

- Hard to overstate how important and useful simulations are in statistics

- Basic idea is to generate repeated samples under a process you design
  - Define a data generating mechanism (e.g. a Normal distribution)
  - Draw a random sample from that data generating mechanism
  - Analyze the sample (e.g. compute the sample mean)
  - Repeat

Simulation

- Hard to overstate how important and useful simulations are in statistics

- Basic idea is to generate repeated samples under a process you design
  - Define a data generating mechanism (e.g. a Normal distribution)
  - Draw a random sample from that data generating mechanism
  - Analyze the sample (e.g. compute the sample mean)
  - Repeat

- Might vary the underlying process to inspect changes
  - Different sample size
  - Different covariate effect
  - Etc

Coding a simulation

- Simulations are natural in the context of iteration

- Write a function (or functions) to:
  - Define data generating mechanism
  - Draw a sample
  - Analyze the sample
  - Return object of interest
- Use a loop / loop function to repeat many times

- Inspect the properties of your analysis …

Coding a simulation

- Simulations are natural in the context of iteration

- Write a function (or functions) to:
  - Define data generating mechanism
  - Draw a sample
  - Analyze the sample
  - Return object of interest
- Use a loop / loop function to repeat many times

- Inspect the properties of your analysis …

…under repeated sampling!!!

Coding a simulation

the context of iteration

ons) to:

g mechanism

est

to repeat many times

- Inspect the properties of your analysis …

…under repeated sampling!!!