Teaching demo: Intro to simulation

Ciaran Evans

Context

- This lecture has been used as the first lecture in STA 279 Statistical Computing
- Students in Statistical Computing have taken STA 112 and are familiar with some R fundamentals; no other background is assumed
- First unit of Statistical Computing is on simulation studies
    - Allows review of some R basics
    - Provides context and motivation for fundamental computing concepts: data types, iteration, good coding practices

Today's lesson

- **Learning goal:** by the end of this lesson, students will be able to implement a short simulation to answer a probability question
- Topics reviewed or introduced:
  - Planning simulations
  - Vectors in R
  - Iteration
- Student participation:
  - Short neighbor/group discussions
  - Dialogue and questions throughout
  - Your turn: activity at the end of the lesson

Warm-up question

Problem: 10 people are at a party, and all of them are wearing hats. They each place their hat in a pile; when they leave, they choose a hat at random. What is the probability at least one person selected the correct hat?

Question: Work with your neighbor to discuss the following question:

- Without calculating probabilities, how could you design an experiment to estimate this probability?

Designing an experiment

Code ideas

Step 1: need 10 hats!

Person 1	person 2	...	Person 10
1	2	...	10

#s 1, 2, ..., 10 (in a vector)

Step 2: randomly assign / shuffle hats

person 1	Person 2	...	Person 10
3	2	...	7

randomly sample (without replacement)

Step 3: check # of people who got their original hat

Step 4: repeat many times!

for loop

Step 1: representing the hats

an object called "hats"

```
hats <- 1:10

hats
```

p.1 p.2 p.3 - - - -

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
hats[3]
```

3rd entry

```
## [1] 3
```

← hat 3

- `hats` is a **vector**, containing the numbers 1 to 10
- entries in a vector are accessed by their index

Step 2: everyone draws a random hat

take a random sample from

object to sample

sample size

```r
hats <- 1:10
randomized_hats <- sample(hats, size = 10,
                          replace = FALSE)
```

without replacement

```r
hats
```

```
## [1]  1  2  3  4  5  6  7  8  9 10
```

```r
randomized_hats
```

p.1 p.2

```
## [1] 10  3  7  4  2  6  5  9  8  1
```

- The `sample` function creates a random sample from a vector
- How many people selected their original hat? 2 people!

Step 3: check who got their original hat

```r
hats
```

```
## [1]  1  2  3  4  5  6  7  8  9 10
```

```r
randomized_hats
```

```
## [1] 10  3  7  4  2  6  5  9  8  1
```

Step 3: check who got their original hat

```
hats
```

```
## [1]  1  2  3  4  5  6  7  8  9 10
```

```
randomized_hats
```

? ? ? ?

```
## [1] 10  3  7  4  2  6  5  9  8  1
```

```
hats == randomized_hats
```

are the entries of hats equal to the entries of randomized_hats?

```
FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
```

== test for equality

= assignment (similar to <-)

Step 3: check who got their original hat

```
hats
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
randomized_hats
```

```
##  [1] 10  3  7  4  2  6  5  9  8  1
```

```
hats == randomized_hats
```

```
   0     0     0     1     0     1     0     0     0     0
FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
```

```
# TRUE is 1, FALSE is 0
sum(hats == randomized_hats)
```

```
## [1] 2
```

2 people got their hat back!

Step 3: check who got their original hat

```
hats
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
randomized_hats
```

```
##  [1] 10  3  7  4  2  6  5  9  8  1
```

```
hats == randomized_hats
```

```
FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
```

```
# TRUE is 1, FALSE is 0
sum(hats == randomized_hats)
```

```
## [1] 2
```

```
# did at least one person get their hat?
sum(hats == randomized_hats) > 0
```

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

at least one person got their hat back!

Code so far

reproducibility ; setting a seed

set.seed(3)

```
hats <- 1:10
randomized_hats <- sample(hats, size = 10,
                          replace = FALSE)
```

```
sum(hats == randomized_hats) > 0
```

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

- In this case, at least one person received their original hat!
- Is this a good estimate of the *probability*?

Step 4: iteration

A `for` loop repeats code many times:

```
nsim <- 10000 # number of simulations
for(i in 1:nsim){
        "repeat the following chunk of code
            for index i = 1, 2, 3, ...., nsim"
}
```

Loop example

start at index i=1

go to index i=5

```
for(i in 1:5){
  print(3)
}
```

```
## [1] 3
## [1] 3
## [1] 3
## [1] 3
## [1] 3
```

5 times

i=1:

print(3)

i=2:

print(3)

⋮

i=5:

print(3)

Loop example

```r
for(i in 1:5){
  print(i)
}
```

Loop example

note: code chunk in our loop
can depend on our index

```r
for(i in 1:5){
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

i=1:
print(1)
i=2:
print(2)
⋮

Step 4: iteration

A `for` loop repeats code many times:

```
nsim <- 10000 # number of simulations
hats <- 1:10

for(i in 1:nsim){
  randomized_hats <- sample(hats, size = 10,
                            replace = FALSE)
  print(sum(hats == randomized_hats) > 0)
}
```

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

Step 4: iteration

initial: results NA NA .... NA

i=1: results T NA ... NA

A for loop repeats code many times: i=2 results T T NA .. NA

repeat NA nsim

```
nsim <- 10000 # number of simulations
hats <- 1:10
results <- rep(NA, nsim) # vector to store results
```

create a vector called "results"

```
for(i in 1:nsim){
  randomized_hats <- sample(hats, size = 10,
                            replace = FALSE)
  results[i] <- sum(hats == randomized_hats) > 0
}
```

storing ith repetition in ith entry of results vector

```
head(results)
```

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

Step 4: iteration

exact (analytical) probability:

0.632120S

A `for` loop repeats code many times:

```
nsim <- 10000 # number of simulations
hats <- 1:10
results <- rep(NA, nsim) # vector to store results

for(i in 1:nsim){
  randomized_hats <- sample(hats, size = 10,
                            replace = FALSE)
  results[i] <- sum(hats == randomized_hats) > 0
}
```

fraction of times result was TRUE

✓

```
mean(results)
```

recall T is 1

F is 0

```
## [1] 0.6307
```

Class activity

For the remainder of class, work with a neighbor on the class activity (link below and on the course website):

https://sta279-example.github.io/class_activities/ca_lecture_1.html

What comes next?

- Continuing probability simulations (gambler's ruin, airplane seating, Monty Hall problem, etc.)
  - setting seeds
  - good coding practices
  - `for` and `while` loops
  - nested loops
  - `if...else if...else` statements
- Statistical simulations
  - answering questions about linear regression models (e.g., does constant variance matter?)
  - ADEMP framework[1]
  - introduction to writing functions

[1] "Using simulation studies to evaluate statistical methods" (Morris *et al.* 2019)

What is the exact probability?