Lecture 1: Intro to simulation

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

← represent hats w/ #s 1-10

Step 1: need 10 hats! (each belongs to one person)

person 1 person 2 3 ... Person 10

Hat 1 ⊓ Hat 2 ⊓ 3 ⊓ ... Hat 10 ⊓

Step 2: randomly "shuffle" hats / ← sampling in R

randomly assign a hat to each person

⊓ 10 ⊓ 4 ⊓ 3 ... ⊓ 9

Step 3: who got their original hat?

Step 4: Repeat many times!

for loop!

Step 1: representing the hats

vector called "hats"

```r
hats <- 1:10

hats
```

← vector 1,2,3,....,10

store the right hand side in the left hand side

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

```r
hats[3]
```

```
[1] 3
```

third entry

- `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

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

hats
```

← vector

1, 2, ..., 10

# of samples

sample

without replacement

vector

take a random sample

object we sample from

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

```
randomized_hats
```

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

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

```r
hats
```

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

```r
randomized_hats
```

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

```r
hats == randomized_hats
```

== test for equality (done for each element)

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

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

how many people got their original hat?

```
[1] 2
```

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

```
[1] TRUE
```

(at least one person got their original hat)

Code so far

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

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

```
[1] TRUE
```

- Is this a good estimate of the probability?

No! Need to repeat many times

Step 4: iteration

A `for` loop repeats code many times:

```r
nsim <- 10000 # number of simulations
for(i in 1:nsim){


}
```

$\underbrace{\texttt{for(i in 1:nsim)}}$ "repeat the following code for $i=1, \dots, nsim$"

$i=1$

(run code)

$i=2$

(run code)

$i=3$

(run code)

$\vdots$

results <- rep(NA, nsim)

NA NA NA . . . . . . NA (10000 times)

Step 4: iteration

A `for` loop repeats code many times:

```r
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
}

head(results)
```

(hats 1, . . . , 10)

"repeat NA nsim times"

results vector (store what happens)

store results

check whether each person got their original hat

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

NA NA NA NA . . . . NA

i = 1 :

(check)

FALSE NA NA NA . . . .

i = 2 :

(check)

FALSE TRUE NA NA . . . .

i = 10000

prob. = proportion of TRUE in results vector

Step 4: iteration

A `for` loop repeats code many times:

```r
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
}

mean(results)
```

magic numbers

```
[1] 0.6357
```

P(at least one person receives original hat) ≈ 0.636

- What if I wanted to repeat the simulation, with a different number of people?

Removing magic numbers

Without magic numbers:

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

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

mean(results)
```

```
[1] 0.6269
```

- Why did I get different results?

Final code

seed

setting a seed: each time I run a process w/ "random" functions, I get the same result

```r
set.seed(3) # set a seed for reproducibility

M <- 10 # number of people at the party
hats <- 1:M # numbered hats
nsim <- 10000 # number of simulations
results <- rep(NA, nsim) # vector to store the results

for(i in 1:nsim){
  # hats are randomly assigned to each person
  randomized_hats <- sample(hats, M, replace = F)

  # did at least one person get their hat back?
  results[i] <- sum(randomized_hats == hats) > 0
}

mean(results)
```

Summary of coding practices

- avoid magic numbers
- set a seed for reproducibility
- use meaningful names
- add comments

Class activity

Work with a neighbor on the class activity (link below and on the course website):

https://sta279-s24.github.io/class_activities/ca_lecture_1.html