Iteration and 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

In R:

Step 1: need 10 hats! (vector #s 1-10)

Person 1 Person 2 . . . .

Hat 1 Π Hat 2 Π

Step 2: randomly "shuffle" hats (Sampling in R)

(randomly assign a hat to each person)

$\frac{\Pi}{10}$ $\frac{\Pi}{4}$ $\frac{\Pi}{3}$ . . . . $\frac{\Pi}{9}$

Step 3: who got their original hat? (compare original #s to new ordering)

Step 4: Repeat many times!

for loop!

Step 1: representing the hats

```r
hats <- 1:10

hats
```

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

```r
hats[3]
```

```
[1] 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

```
1 hats <- 1:10
2 randomized_hats <- sample(hats, size = 10, replace = FALSE)
3
4 hats
```

items to sample

object to sample from

without replacement (people can't share a hat)

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

```
1 randomized_hats
```

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

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

idea: check if `randomized_hats[1] == 1`, `randomized_hats[2] == 2`, ⋮

implementation: `randomized_hats == hats`

Step 3: check who got their original hat

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

```
hats
```

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

```
randomized_hats
```

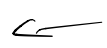
```
[1]  4  7  1  3  8 10  9  2  5  6
```

```
hats == randomized_hats
```

test for equality (done for each element)

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

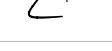
```
# TRUE is 1, FALSE is 0
sum(hats == randomized_hats)
```



how many got their original hat

```
[1] 0
```

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



did at least one person get their original hat?

```
[1] FALSE
```

Code so far

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

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

repeat these two lines many times

```
[1] FALSE
```

- Is this a good estimate of the probability?

No! Need to repeat

Step 4: iteration

A `for` loop repeats code many times:

```
1 nsim <- 10000 # number of simulations
2 for(i in 1:nsim){
3
4
5 }
```

"repeat the following code for i=1, i=2, ..., i=nsim"

results : NA NA NA ... NA

Step 4: iteration

A `for` loop repeats code many times:

```
 1 nsim <- 10000 # number of simulations
 2 hats <- 1:10
 3 results <- rep(NA, nsim) # vector to store results
 4
 5 for(i in 1:nsim){
 6   randomized_hats <- sample(hats, size = 10, replace = FALSE)
 7   results[i] <- sum(hats == randomized_hats) > 0
 8 }
 9
10 head(results)
```

"repeat NA nSim times"

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

i = 1 : (check)

TRUE NA NA NA ... NA

i = 2 : (check)

TRUE TRUE NA NA ... NA

i = 10000

Step 4: iteration

A `for` loop repeats code many times:

```
 1 nsim <- 10000 # number of simulations
 2 hats <- 1:10
 3 results <- rep(NA, nsim) # vector to store results
 4
 5 for(i in 1:nsim){
 6   randomized_hats <- sample(hats, size = 10, replace = FALSE)
 7   results[i] <- sum(hats == randomized_hats) > 0
 8 }
 9
10 mean(results)
```

or: `sum(results) / nsim`

```
[1] 0.6231
```

P(at least one person receives original hat) ≈ 0.623

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

Removing magic numbers

Without magic numbers:

```
 1 nsim <- 10000 # number of simulations
 2 n_people <- 10 # number of people
 3 hats <- 1:n_people
 4 results <- rep(NA, nsim) # vector to store results
 5
 6 for(i in 1:nsim){
 7   randomized_hats <- sample(hats,
 8                             size = n_people,
 9                             replace = FALSE)
10   results[i] <- sum(hats ==
11                       randomized_hats) > 0
12 }
13
14 mean(results)
```

```
[1] 0.6316
```

- Why did I get different results?

Final code

each time I run a process w/ "random" functions, I get same result

```
1  set.seed(3) # set a seed for reproducibility
2
3  n_people <- 10 # number of people at the party
4  hats <- 1:n_people # numbered hats
5  nsim <- 10000 # number of simulations
6  results <- rep(NA, nsim) # vector to store the results
7
8  for(i in 1:nsim){
9    # hats are randomly assigned to each person
10   randomized_hats <- sample(hats, n_people, replace = F)
11
12   # did at least one person get their hat back?
13   results[i] <- sum(randomized_hats == hats) > 0
14 }
15
16 mean(results)
```

Map?

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

n_people <- 10 # number of people at the party
hats <- 1:n_people # 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, n_people, replace = F)

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

mean(results)
```

How could we do this with `map` instead?

Need to write a function!

Map: writing a function

```r
set.seed(3)

hat_match <- function(n){
  hats <- 1:n
  randomized_hats <- sample(hats, n, replace = F)
  sum(randomized_hats == hats) > 0
}

hat_match(10)
```

performs inner part of for loop

```
[1] FALSE
```

```r
hat_match(10)
```

```
[1] FALSE
```

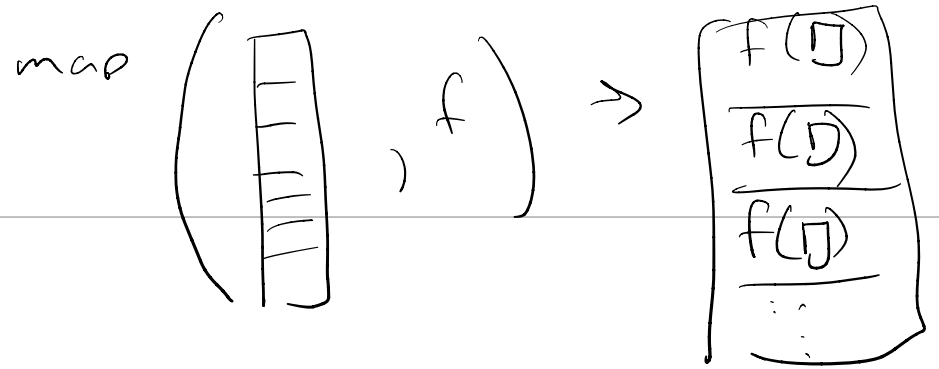
```r
hat_match(10)
```

```
[1] FALSE
```

```r
hat_match(10)
```

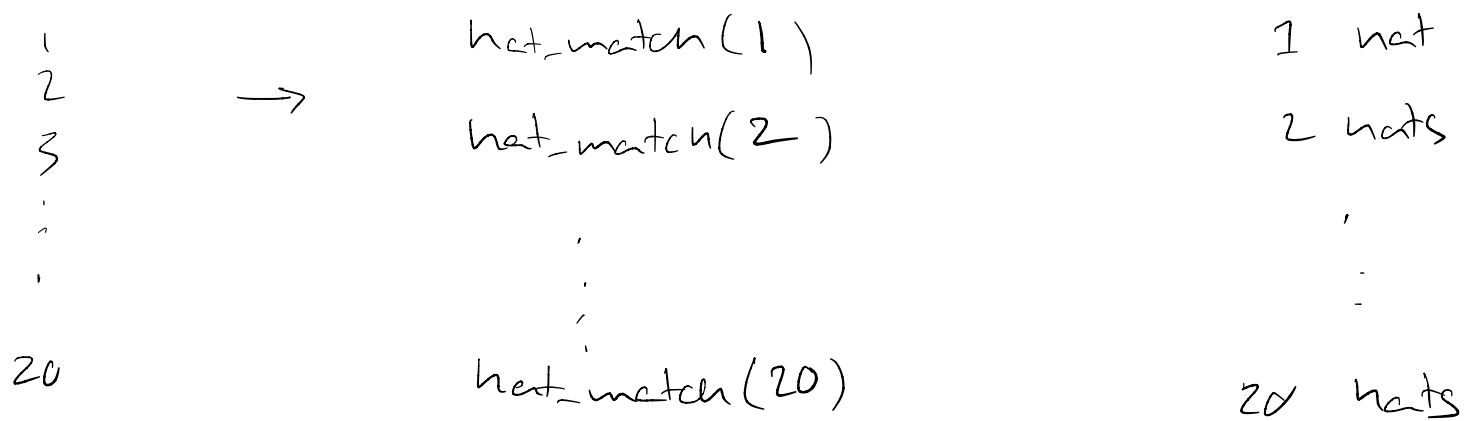
```
[1] TRUE
```

Map: iterating




```
 1 set.seed(3)
 2 nsim <- 20
 3
 4 hat_match <- function(n){
 5   hats <- 1:n
 6   randomized_hats <- sample(hats, n, replace = F)
 7   sum(randomized_hats == hats) > 0
 8 }
 9
10 map(1:nsim, hat_match)
```

Will this do what I want?

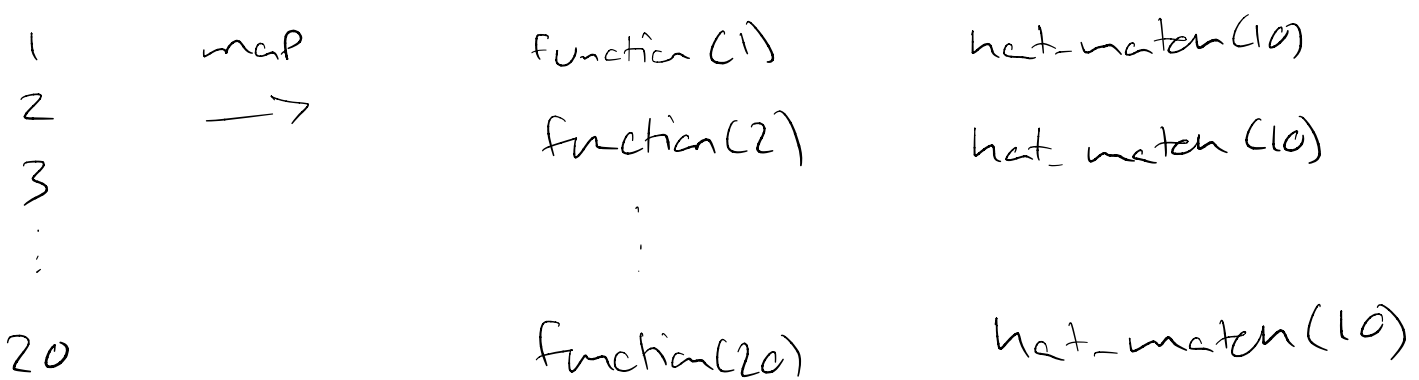


Map: iterating

```
 1 set.seed(3)
 2 nsim <- 20
 3 n_people <- 10
 4
 5 hat_match <- function(n){
 6   hats <- 1:n
 7   randomized_hats <- sample(hats, n, replace = F)
 8   sum(randomized_hats == hats) > 0
 9 }
10
11 map_lgl(1:nsim, function(i) hat_match(n_people))
```

vector of booleans

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



Map: iterating

```r
set.seed(3)
nsim <- 1000
n_people <- 10

hat_match <- function(n){
  hats <- 1:n
  randomized_hats <- sample(hats, n, replace = F)
  sum(randomized_hats == hats) > 0
}

map_lgl(1:nsim, function(i) hat_match(n_people)) |>
  mean()
```

```
[1] 0.605
```

Class activity

https://sta279-f25.github.io/class_activities/ca_12.html

- Work with a neighbor on the class activity
- At the end of class, submit your work as an HTML file on Canvas (one per group, list all your names)