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

Step 1: representing the hats

```
1  hats <- 1:10
2
3  hats
```

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

```
1  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

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

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

```r
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?

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

```r
hats == randomized_hats
```

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

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

```
[1] 0
```

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

```
[1] FALSE
```

Code so far

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

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

```
[1] FALSE
```

- Is this a good estimate of the probability?

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  }
```

Step 4: iteration

A `for` loop repeats code many times:

```r
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)
```

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

Step 4: iteration

A `for` loop repeats code many times:

```r
 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)
```

```
[1] 0.6231
```

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

Removing magic numbers

Without magic numbers:

```r
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

```r
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?

```r
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?

Map: writing a function

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

```
[1] FALSE
```

```r
1  hat_match(10)
```

```
[1] FALSE
```

```r
1  hat_match(10)
```

```
[1] FALSE
```

```r
1  hat_match(10)
```

```
[1] TRUE
```

Map: iterating

```r
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

```r
set.seed(3)
nsim <- 20
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))
```

```
 [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)