Iteration and simulation

Class activity

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

- Work independently or 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)

Warm-up question

- A roulette wheel has 38 slots numbered 00, 0, and 1–36. Two are green, 18 are red, and 18 are black.

- If a gambler bets based on color, the return on a $1 bet is $2

- A gambler has $50, and will continuously bet $1 on red until they double their money (have $100) or lose the money they came with

- What is the probability the gambler doubles their money?

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

Designing an experiment

Step 1 :   need a   roulette wheel
             (38 slots,   2g, 18r, 18b)

vector : $\underbrace{g,g}_{2}, \underbrace{r, ..., r}_{18}, \underbrace{b, ..., b}_{18}$

money :   starts at $0

Step 2 :   spin the   wheel , check outcome

Step 3 :   update   money
             if spin is   red :   money = money +1
             if spin is not red :   money = money -1

Step 4 :   keep spinning   until   money = 100   or money = 0

Step 5 :   repeat   the   whole   process many times

Step 1: representing the roulette wheel

```
1  wheel <- c(rep("green", 2), rep("black", 18), rep("red", 18))
2
3  wheel
```

2g *18 b* *18 r*

```
 [1] "green" "green" "black" "black" "black" "black" "black" "black"
"black"
[10] "black" "black" "black" "black" "black" "black" "black" "black"
"black"
[19] "black" "black" "red"   "red"   "red"   "red"   "red"   "red"
"red"
[28] "red"   "red"   "red"   "red"   "red"   "red"   "red"   "red"
"red"
[37] "red"   "red"
```

- rep repeats a value a specified number of times

- c() combines vectors into a single vector

Step 2: spin the wheel!

```
1  spin <- sample(wheel, size = 1)
2
3  spin
```

[1] "black"

if   spin   is red:

money = money +1

otherwise:

money = money -1

Step 3: change in money

```r
 1  money <- 50
 2  spin <- sample(wheel, size = 1)
 3
 4  if(spin == "red"){
 5    money <- money + 1
 6  } else {
 7    money <- money - 1
 8  }
 9
10  spin
```

```
[1] "red"
```

```r
 1  money
```

```
[1] 51
```

- if the result was red, gain a dollar

- otherwise, lose a dollar

Step 3: change in money

Another way of writing the conditional statement:

```
1  money <- 50
2  spin <- sample(wheel, size = 1)
3
4  money <- ifelse(spin == "red", money + 1, money - 1)
5
6  spin
```

Condition to check

if condition is true

otherwise

(condition is false)

[1] "black"

```
1  money
```

[1] 49

Step 4: keep spinning

The gambler continues to bet until they have $0 or $100.

Question: Is a `for` loop appropriate for iterating the betting process?

for loop : repeats a chunk of code a fixed #
of times

while loop : repeats code while a condition is
true

Step 4: keep spinning

```
1  money <- 50 # starting money
2
3  while(money > 0 & money < 100){
4    spin <- sample(wheel, size = 1)
5    money <- ifelse(spin == "red", money + 1, money - 1)
6  }
7
8  money
```

repeat   until   either   money ==0
or money ==100

[1] 0

- `while` loop: repeat the process until the condition is true

Step 5: repeat the process

```r
1   set.seed(279)
2   nsim <- 1000        ← # Simulations
3   results <- rep(NA, nsim)   ← vector to store results
4
5   for(i in 1:nsim){    ← # times we play the game
6     money <- 50 # starting money
7
8     while(money > 0 & money < 100){
9       spin <- sample(wheel, size = 1)
10      money <- ifelse(spin == "red", money + 1, money - 1)
11    }
12
13    results[i] <- ...money == 100
14  }
```

- What should I check at each iteration?

results : vector of TRUES and FALSES

\# TRUES = # times got to 100

mean(results)   or   sum(results)/nsim

Step 5: repeat the process

```r
set.seed(279)
nsim <- 1000
results <- rep(NA, nsim)

for(i in 1:nsim){
  money <- 50 # starting money

  while(money > 0 & money < 100){
    spin <- sample(wheel, size = 1)
    money <- ifelse(spin == "red", money + 1, money - 1)
  }

  results[i] <- money == 100
}

mean(results)
```

```
[1] 0.008
```