Simulating probabilities in R

SSI - 13200

January 22, 2019

Exercise

- If we have a deck of cards, and we draw 4 cards randomly (without replacement), what is the probability that at least 1 of those is an ace?

Mathematic Solution

- As we discussed last time, this is easier to do using the complement

```
math=1-((48/52)*(47/51)*(46/50)*(45/49))
math
```

```
## [1] 0.2812633
```

Using Simulations in R

- We can use simulation to estimate this. How can we transform this problem into a series of programming steps?

Step 1: Simulate one deck

```r
deck <- rep(1:13, 4)
deck
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13  1  2  3  4
## [24] 11 12 13  1  2  3  4  5  6  7  8  9 10 11 12 13  1
## [47]  8  9 10 11 12 13
```

Step 2: Randomly draw 4 cards

```r
draw <- sample(deck, 4, replace = FALSE)
draw
```

```
## [1] 11  3  1  9
```

Step 3: Capture the relevant information

```r
atleast1 <- ifelse(1 %in% draw, 1, 0)
atleast1
```

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

Step 4: Repeat so many times!

-For this, we will introduce loops

```
deck <- rep(1:13, 4)
atleast1 <- rep(NA, 10000)

for (i in 1:10000) {
    draw <- sample(deck, 4, replace = FALSE)
    atleast1[i] <- ifelse(1 %in% draw, 1, 0)
}

mean(atleast1)
```

```
## [1] 0.2901
```

Introduction to the De Montmort Problem

- Imagine you have a pile of the 13 spades cards and another of 13 hearts
- Assume they are well shuffled, and you turn over one from each pile at the same time
- What is the probability that none of the cards match?

To do this in R

- We need a “pile” of hearts and a “pile” of spades

```
hearts <- sample(1:13, 13)
spades <- sample(1:13, 13)
hearts
```

```
##  [1]  8 12  7  4 10  3  9 13  1  5 11  6  2
```

```
spades
```

```
##  [1] 10  2 12  3 13  4  8  7 11  6  1  5  9
```

Create an empty “success” vector

```r
correct <- rep(NA,13)
correct
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA
```

Test if they are the same

```r
for (i in 1:13) {
  correct[i] <- ifelse(hearts[i] == spades[i], 1, 0)
  }
correct
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Sum the number of correct times

```
n_correct <- sum(correct)
n_correct
```

```
## [1] 0
```

Before we can do this repeatedly...

- We will store the number of correct times in a vector n_correct:

```r
n_correct <- rep(NA, 10000)
n_correct[1:100]
```

```
##   [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
##  [24] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
##  [47] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
##  [70] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
##  [93] NA NA NA NA NA NA NA NA
```

To do this repeatedly…

- We need a loop within a loop

```
for (f in 1:10000) {
  hearts <- sample(1:13, 13)
  spades <- sample(1:13, 13)
  correct <- rep(NA,13)

    for (i in 1:13) {
      correct[i] <- ifelse(hearts[i] == spades[i], 1, 0)
    }
  n_correct[f] <- sum(correct)
}
```

How many times did they match?

```
table(n_correct)
```

```
## n_correct
##    0    1    2    3    4    5    6 
## 3638 3818 1793  550  160   35    6
```

What percentage of times did we get 0 correct?

```r
table(n_correct)[1]/10000
```

```
##      0
## 0.3638
```

- That is an approximation of the probability of getting 0 correct

Mathematical Solution

- The math is more complicated than we need for this class
- The actual solution is .3679

Expected Value

- How many cards should we expect to match?

```r
mean(n_correct)
```

```
## [1] 0.9905
```

Exercise

After shuffling a deck of cards, what is the probability that none are in the same place they were before? How many should we expect to be in the same place they were before?