Lecture 8: Probability and Simulation

Ingmar Sturm
UCSB
2024-07-08

Special thanks to Robin Liu for select course content used with permission.

Probability

Rolling two dice

Roll two fair dice, one red and one green.

What is the *probability* that their sum equals 6?

A simulation-based answer.

1. Roll the dice
2. Sum up the two numbers.
3. Does the sum add up to 6? Write down TRUE or FALSE.
4. Write this data as a line in a notebook
5. Repeat a **ton** of times.

The probability is the proportion of true responses to step 3.

Rolling two dice

rep		red	green	red+green	red+green==6?
1		4	3	7	FALSE
2		4	2	6	TRUE
3		1	4	5	FALSE
4		2	6	8	FALSE
5		6	1	7	FALSE

In 5 replications, we found a proportion of $\frac{1}{5}$ of true responses. The *probability* is this proportion when we take $n \to \infty$ replications.

You might know already the answer is $\frac{5}{36}$ (1 + 5, 2 + 4, 3 + 3, 4 + 2, 5 + 1). The above connects this theoretical answer to real life experiments; it is the **frequentist** interpretation of probability.

Since this is a coding class (and not a probability theory class), we will adopt this simulation-based "definition" and interpretation of probability.

Intuitive "definition" of probability

An **experiment** is repeatable with random outcomes. We imagine performing the experiment a **huge** number of times.

- Roll two fair dice and add the results

An **event** is a logical (TRUE or FALSE) outcome of an experiment.

- Does the sum equal 6? Either TRUE or FALSE.

The **probability** of an event is the proportion of TRUE outcomes after a **huge** number of replications of the experiment.

We will use R to simulate this experiment and determine the probability.

Setting a seed

Demonstrate pseudorandomness

```r
sample(1:10)
```

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

```r
set.seed(100)
sample(1:10)
```

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

Simulating the two dice

The `sample` function returns a vector of random elements from an input vector.

```r
# the second argument is the number of elements to return.
sample(c("cat", "fish", "llama", "hedgehog"), 1)
```

```
## [1] "fish"
```

```r
sample(c("cat", "fish", "llama", "hedgehog"), 2)
```

```
## [1] "hedgehog" "llama"
```

We can represent rolling a fair die as follows:

```r
sample(1:6, 1)
```

```
## [1] 6
```

Simulating the two dice

Roll two dice, one green, one red. What is the probability the sum equals six? Simulate using $10,000$ replications.

```r
nreps <- 10000
count <- 0

for (rep in seq_len(nreps)) {
  green <- sample(1:6, 1)
  red <- sample(1:6, 1)
  dice_sum <- green + red
  if (dice_sum == 6) {
    count <- count + 1 # The event occurs, so we increment the counter
  }
}

count/nreps
```

```
## [1] 0.1396
```

Our answer is close to the theoretical value of $\frac{5}{36} \approx 0.1389$.

Simulating dice rolls

Lets generalize to rolling d fair, six-sided dice and find the probability their sum equals k.

It will be convenient to `sample` multiple rolls at once.

```r
sample(1:6, 7, replace = T)
```

```
## [1] 5 4 2 4 3 5 3
```

The `replace = T` indicates sampling *with replacement*. It is FALSE by default. Without this, we would not be able to simulate rolling the same number twice.

Try running the following line of code:

```r
sample(1:6, 7) # replace = F by default
```

Simulating dice rolls

Roll 7 fair six-sided dice. What is the probability their sum equals 20? Simulate using $10,000$ replications.

```r
d <- 7
k <- 20
nreps <- 10000

count <- 0 # Count the number of TRUE outcomes

for (rep in seq_len(nreps)) {
  rolls <- sample(1:6, d, replace = T)
  if (sum(rolls) == k) {
    count <- count + 1 # The event occurs, so we increment the counter
  }
}

count/nreps
```

```
## [1] 0.0546
```

The above shows the probability that the sum of 7 dice equals 20 is roughly 0.055. Harder to compute theoretically!

Simulating dice rolls

We can further improve this code:

```r
d <- 7
k <- 20
nreps <- 10000
sums <- vector(length = nreps) # Store the sums in each replication

for (rep in seq_len(nreps)) {
  rolls <- sample(1:6, d, replace = T)
  sums[rep] <- sum(rolls)
}
mean(sums == k) # !!
```

```
## [1] 0.0561
```

We got rid of the `count` variable. Take a moment to think about what `mean(sums == k)` is doing.

01:00

Simulating dice rolls

A final improvement with the `replicate` function.

```
replicate(10000, sum(sample(1:6, size =  7, replace = T))) |> head(10)
```

```
##  [1] 31 24 24 23 26 24 21 23 26 22
```

Performs 10 replications of the provided expression.

```
sums <- replicate(10000, sum(sample(1:6, size = 7, replace = T)))
mean(sums == 20)
```

```
## [1] 0.0553
```

We moved the replication to the `replicate` function and no longer need a loop. Then we used recycling and vectorization to find the desired proportion.

The key idea: create a logical vector containing, for each replication, TRUE if the event occurs and FALSE if the event does not occur. Then take the mean of this vector. This estimates the probability of the event.

Your turn

Roll three fair six-sided dice. What's the probability that the minimum of the scores is greater than 2?

Hint: Consider using the `min` function:

```
min(2, -6, 700, -1)
```

```
## [1] -6
```

05:00

More dice rolling

A pair of fair dice is rolled in order. What is the probability that the second die lands on a higher value than the first die?

05:00

Derangement

A **derangement** is a permutation of of numbers $1, 2, \ldots, n$ such that no number is in its original position.

Example: if $n = 5$,

- 2, 1, 5, 3, 4 is a derangement
- 2, 1, 5, 4, 3 is not a derangement; 4 is in the 4th position.

Let $n = 100$ and suppose we picked a permutation of numbers 1:100 randomly. What is the probability of a derangement? Simulate drawing a permutation `nrep=10^4` times.

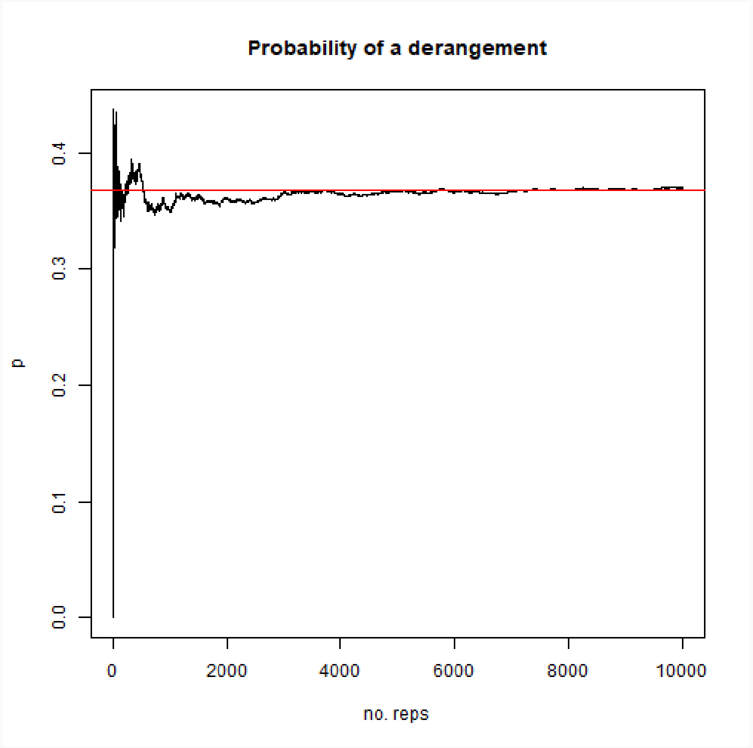
Hint: `sample(1:100)` gives a random permutation of numbers $1, 2, \ldots, 100$.

08:00

Derangement

Amazingly, it can be shown for large n that the probability of a derangement is approximately $1/e \approx 0.368$.

Create the following plot:



Expectation

Recall our notebook of experimental results.

rep		red	green	red+green	red+green==6?
1		4	3	7	FALSE
2		4	2	6	TRUE
3		1	4	5	FALSE

The columns **red**, **green**, and **red+green** give numerical results of each experiment (we will soon call them random variables).

The **expectation** or **expected value** of a column is the long-run *average* of all the entries in the column.

$$\lim_{n \to \infty} \frac{x_1 + x_2 + \ldots + x_n}{n}.$$

We approximate the expectation by performing the experiment a **huge** number of times and taking the average.

Expectation

rep		red	green	red+green	red+green==6?
1		4	3	7	FALSE
2		4	2	6	TRUE
3		1	4	5	FALSE
4		2	6	8	FALSE
5		6	1	7	FALSE

After 5 replications, what is the estimated expectation of the sum of scores on the red and green dice?

```
mean(c(7, 6, 5, 8, 7))
```

```
## [1] 6.6
```

It can be shown the theoretical value is 7.

Expectation

We approximate the expectation by performing the experiment a **huge** number of times and taking the average.

Estimate the expectation of the sum of two dice using $1,000$ replications:

```r
sample(1:6, size = 2, replace = T)
```

```
## [1] 2 3
```

```r
result <- replicate(1000, sum(sample(1:6, size = 2, replace = T)))
mean(result)
```

```
## [1] 6.854
```

We should get closer to the theoretical as we do more replications.

Expectation

Consider the following experiment.

1. Roll a fair six-sided die 100 times.
2. Count the number of rolls before the first score of 6 was rolled.

What is the expected number of rolls before a 6 was seen? Estimate the expectation using 10,000 replications.

08:00