

Explaining the Birthday Problem

Fernando Hoces de la Guardia

09/03/2021

Birthday problem

What is the probability that at least two people share the same birthday in this room?

Is is $\frac{1}{365} \times 17 = 0.0465753$

Analytical solution

But actually when we compute the math. We get an surprising result:

$$\begin{aligned} 1 - \bar{p}(n) &= 1 \times \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \times \cdots \times \left(1 - \frac{n-1}{365}\right) \\ &= \frac{365 \times 364 \times \cdots \times (365 - n + 1)}{365^n} \\ &= \frac{365!}{365^n (365 - n)!} = \frac{n! \cdot \binom{365}{n}}{365^n} \\ p(n = 17) &= 0.315 \end{aligned} \tag{1}$$

Simulations

- 1 - Simulate 10,000 rooms with $n = 17$ random birthdays, and store the results in matrix where each row represents a room.
- 2 - For each room (row) compute the number of unique birthdays.
- 3 - Compute the average number of times a room has 17 unique birthdays, across 10,000 simulations, and report the complement.

```
birthday.prob = function(n.pers_var, n.sims_var) {  
  # simulate birthdays  
  birthdays = matrix(round(runif(n = n.pers_var * n.sims_var, min = 1, max = 365) ),  
                      nrow = n.sims_var, ncol = n.pers_var)  
  # for each room (row) get unique birthdays  
  unique.birthdays = apply(birthdays, 1,  
                           function(x) length( unique(x) ) )  
  # Indicator with 1 if all are unique birthdays  
  all.different = 1 * (unique.birthdays==n.pers_var)  
  # Compute average time all have different birthdays  
  result = 1 - mean(all.different)  
  return(result)  
}  
  
bp_sim = birthday.prob(n.pers_var = 17, n.sims_var = 10000)  
print(bp_sim)
```

```
## [1] 0.3205
```

Results

- Many people originally think of a prob $\sim \frac{1}{365} \times n = 0.047$
- However the true probability is of $p(n = 17) = 0.315$
- And the simulated probability is of 0.3205