question 2 views

Daily Challenge 22.6

(**Due: Monday 2/25 at 12:00 noon Eastern**)

Today I'll ask you to write up a Python simulation that *seems* to have nothing to do with the gamma function.

(1) The coupon collector's problem.

Suppose that you buy one box of cereal every day. Each box of cereal contains a randomly chosen coupon, and there are n different types of coupons in total. On average, how many days will it take you to collect all n coupons?

For instance, if there are 3 coupons, one possible run might look like this:

1. On the first day, I get coupon 1.
2. On the second day, I get coupon 3.
3. On the third day, I get coupon 3 again. Ignore duplicates; I keep going until I have at least one of every coupon.
4. On the fourth day, I get coupon 1. Ignore.
5. On the fifth day, I get coupon 2. Now the game stops because I have all three.

So the above trial would have $n = 3$ and $t = 5$ days. We want to know the *average* number of days t that you need to get n coupons, as a function of n.

Write a Python simulation which takes in a value of n and runs some large number, say $1,000$ trials, then outputs the average time until all n coupons are collected. Your code might look like this:

```
def coupon_collector(n, trials=1000):
    """
    Takes in a number of different coupons, returns average amount of time to collect all of them
    """
    times_list = []

    for trial in range(trials):

    ## Run a simulation that picks a random coupon every day and increments a counter until you have all of them.

    ## You might want to use np.random.choice to pick a random coupon, store the coupons you have so far in a list,
    ## then end the while loop when the list has all n different coupons. Then append the time counter to your list.

    return np.mean(times_list) ## This just takes the average
```

As a check, when there are $n = 50$ coupons, your program should return an average of 225 days.

Now suppose I tell you that the closed-form answer to this question is approximately

$$(\text{average number of days}) \approx n \log n + \gamma n + \frac{1}{2},$$

where γ is some special number, and the approximation becomes better as n gets larger.

Plug in some large value of n to find the left side of the equation above, and then solve for the number γ,

$$\gamma \approx \frac{1}{n}\left((\text{average number of days}) - n \log n - \frac{1}{2}\right).$$

What value of γ do you find? It should be close to 0.577.

Post your code on Github.

daily_challenge

Updated 1 month ago by Christian Ferko

the students' answer, *where students collectively construct a single answer*

The first part of the solution can be found on GitHub; we shall let n = 100 for the sake of a large number that still has sub 1-minute computing times.

$\text{coupon\_collector}(100) = 520.37$

and thus by calculating

$$\frac{1}{100}\left(520.37 - 100 \cdot \log(100) - \frac{1}{2}\right) = 0.593...$$

Close enough to be close.

Updated 1 month ago by Logan Pachulski

the instructors' answer, *where instructors collectively construct a single answer*

Click to start off the wiki answer

followup discussions *for lingering questions and comments*