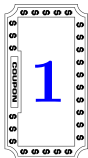


# Coupon collector's problem



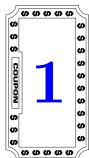
# Coupon collector's problem

There are  $n$  different types of coupons.

Each purchase comes with one random coupon.

Every coupon is equally likely to appear.

How many purchases needed to collect all coupons?



# No definite answer

The answer is **random**:

- If **lucky**, then  $n$  purchases are enough.
- If **unlucky**, then no purchases will be enough.

More suitable question:

How many purchases needed **on average** to collect all coupons?



## Expected value

$X$  is a discrete random variable with real number outcomes  $x_1, \dots, x_k, \dots$

Expected value of  $X$  is

$$E[X] = x_1 P(X = x_1) + \dots + x_k P(X = x_k) + \dots,$$

the value of  $X$  on average if same experiment is repeated over and over again.

**Note:** Assume this infinite sum is well-defined.

## Expected value: example

Play the following coin flipping game:

- Win 1 dollar if coin comes out head,
- Lose 1 dollar if coin comes out tail.

$X$  is the amount of money won from this game.

$$\begin{aligned} E[X] &= 1 \times P(X = 1) + (-1) \times P(X = -1) \\ &= 1 \times 0.5 + (-1) \times 0.5 = 0. \end{aligned}$$



## Geometric random variable

- You perform a sequence of **independent** trials.
- Each trial has **two outcomes**: success or failure.
- Probability of success for every trial is  $p$ .

**Geometric random variable**  $X$  is total number of trials needed until first success.



## Geometric random variable: probabilities

For the first success to be at  $k$ -th trial:

- First  $(k - 1)$  trials are failures,
- The  $k$ -th trial is success.

Probability for this event is

$$\begin{aligned} P(X = k) &= \underbrace{(1 - p) \times \dots \times (1 - p)}_{(k-1) \text{ times}} \times p \\ &= (1 - p)^{k-1} p. \end{aligned}$$



## Geometric random variable: expected value

Expected value  $E[X]$  is equal to

$$P(X = 1) + 2 P(X = 2) + \dots + k P(X = k) + \dots$$

Plugging in the probabilities:

$$\begin{aligned} E[X] &= p + \dots + k (1 - p)^{k-1} p + \dots \\ &= p \sum_{k=1}^{\infty} k (1 - p)^{k-1} = \frac{1}{p}. \end{aligned}$$

**Note:** This is derivative of geometric series.

**Note:** Write  $E[X] = \frac{1}{p}$  on separate piece of paper.



## Back to coupon collector

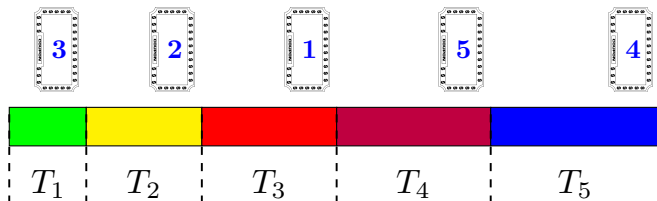
$T$  = number of purchases needed to collect all coupons.

Coupon collector's problem: Compute  $E[T]$ .



## Change of perspective

- $T_1$  = purchases needed to get 1 coupon.
- $T_2$  = purchases needed to get 2 coupons after collecting 1 coupon.
- $T_k$  = purchases needed to get  $k$  coupons after collecting  $k - 1$  coupons.



# Linearity of expectation

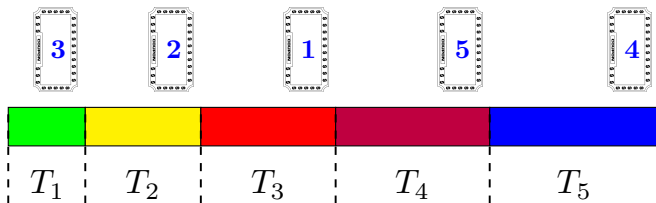
Then to collect all  $n$  coupons, we need

$$T = T_1 + T_2 + \dots + T_n.$$

So

$$E[T] = E[T_1] + E[T_2] + \dots + E[T_n].$$

This is called **linearity** of expectations.



## Geometric random variable comes back

After collecting  $k - 1$  coupons, for next purchase:

- Failure = getting coupon you already have.

There are  $k - 1$  choices.

- Success = getting a new coupon.

There are  $n - k + 1$  choices.

Have:



Don't Have:



## Geometric random variable comes back

Then  $T_k$  is **geometric** random variable with

success probability  $p = \frac{n - k + 1}{n}$ .

$$\text{So } E[T_k] = \frac{1}{p} = \frac{n}{n - k + 1}.$$

## Solution to coupon's collector problem

$$\begin{aligned}E[T] &= E[T_1] + E[T_2] + E[T_3] + \dots + E[T_n] \\&= \frac{n}{n} + \frac{n}{n-1} + \frac{n}{n-2} + \dots + \frac{n}{1} \\&= n \left( \frac{1}{n} + \frac{1}{n-1} + \frac{1}{n-2} + \dots + \frac{1}{1} \right) \\&= n \left( 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} \right).\end{aligned}$$

This is [harmonic series](#), and

$$E[T] \text{ is approximately } n \int_1^n \frac{1}{x} dx = n \log n.$$

## Values of $E[T]$

- Number of coupons  $n = 5$ ;  
Average purchases needed  $E[T] = 11.4$ .
- Number of coupons  $n = 25$ ;  
Average purchases needed  $E[T] = 95.4$ .
- Number of coupons  $n = 100$ ;  
Average purchases needed  $E[T] = 518.7$ .

How close is your guess to the answer?

**THANK YOU!**