

derangements

June 14, 2022

1 Derangements of Sequences

Given a known sequence of distinct elements, we consider perturbations of that sequence, known as *arrangements* or, excluding the original sequence, *rearrangements*. In particular, we seek perturbations such that none of the perturbed elements match the elements in the same position in the original sequence. Such perturbations are known as *derangements*.

1.1 Hat Check Problem

The original problem of derangements was called the *Hat Check Problem*. Consider that N people arrive at some evening venue, one at a time, each wearing a hat. The hats are checked into storage for later collection, but the hat clerk, being a temporary, last-minute replacement, forgets to ticket each hat. Thus, when the people leave the venue later that night in some arbitrary order, they are given a random hat by the clerk. We must further suppose that it is very dark outside the venue and also inside the hat-check area, such that people do not notice they have the wrong (or right) hat until everyone has already collected a hat and left.

The question of interest is this: What is the probability that none of the N people receive their correct hat at the end of the night?

1.1.1 Counting derangements

For convenience, we uniquely label the people in order of their arrival as P_1, P_2, \dots, P_N . Similarly, we uniquely label their respective hats as H_1, H_2, \dots, H_N . Hence, only the ordering H_1, \dots, H_N is correct, and any other ordering (i.e. any of the other $N! - 1$ permutations or rearrangements) is incorrect.

However, we can further distinguish between rearrangements where some (one or more) people get their right hats, and derangements where no-one gets their right hat. Thus, we can ask: How many possible derangements are there?

To count the number, $D(N)$, of derangements, suppose that person P_1 is given the wrong hat H_i , for some $i \in \{2, 3, \dots, N\}$, which can clearly happen in exactly $N - 1$ ways. We might write this as the assignment $P_1 \leftarrow H_i$.

Now, because hat H_i has been given to person P_1 , then person P_i cannot receive their correct hat H_i . Thus, we can now split the problem into two sub-problems, such that either:

1. Person P_i gets hat H_1 , i.e. $P_i \leftarrow H_1$.
2. Person P_i does not get hat H_1 , i.e. $P_i \leftarrow H_j$, with $j \in \{2, \dots, i - 1, i + 1, \dots, N\}$.

For case 1, the partial hats-to-people assignments look like

$$P_1 \cdots P_i \cdots P_N \quad (1)$$

$$\leftarrow H_i \cdots H_1 \cdots \quad (2)$$

Consequently, 2 people so far, namely P_1 and P_i , do not have their correct hats. Thus, a full derangement requires that none of the remaining $N-2$ people receive their correct hats. Specifically, **none** of the following assignments are permitted:

$$P_2 \cdots P_{i-1} P_{i+1} \cdots P_N \quad (3)$$

$$\leftarrow H_2 \cdots H_{i-1} H_{i+1} \cdots H_N \quad (\text{not permitted}) \quad (4)$$

Thus, the remaining $N-2$ assignments may be deranged in exactly $D(N-2)$ ways.

For case 2, we have already established that person P_i cannot be given hat H_1 (since this is just case 1). Likewise, none of the remaining $N-2$ people may receive their correct hats. Thus, a full derangement requires that **none** of the following assignments is permitted:

$$P_2 \cdots P_{i-1} P_i P_{i+1} \cdots P_N \quad (5)$$

$$\leftarrow H_2 \cdots H_{i-1} H_1 H_{i+1} \cdots H_N \quad (\text{not permitted}) \quad (6)$$

Thus, since we have already counted person P_1 , none of the remaining $N-1$ people may be given the hats listed above. Hence, the $N-1$ assignments may be deranged in $D(N-1)$ ways.

Consequently, combining cases 1 and 2, we obtain the recurrence relation

$$D(N) = (N-1) [D(N-2) + D(N-1)] , \quad (7)$$

since we observed above that person P_1 may get the wrong hat in $N-1$ distinct ways.

For boundary conditions, we observe that if there is only $N=1$ person, then the only possible assignment is $P_1 \leftarrow H_1$, and there can be no derangements, i.e. $D(1) = 0$. Likewise, if $N=2$, then the only possible assignments are $P_1 P_2 \leftarrow H_1 H_2$ and $P_1 P_2 \leftarrow H_2 H_1$, of which the first assignment is correct and the second is a derangement, giving $D(2) = 1$. Hence, the recursion holds for $N \geq 2$ if we define $D(0) \doteq 1$ for convenience.

Unrolling the recursion, we observe that

$$D(1) = 0 , \quad (8)$$

$$D(2) = 1 , \quad (9)$$

$$D(3) = 2 [D(2) + D(1)] = 2 \times (1 + 0) = 2! \quad (10)$$

$$D(4) = 3 [D(3) + D(2)] = 3 [2! + 1] = 3! + 3 \quad (11)$$

$$D(5) = 4 [D(4) + D(3)] = 4 [(3! + 3) + (2!)] = 4! + 4 \times 5 \quad (12)$$

$$D(6) = 5 [D(5) + D(4)] = 5 [(4! + 20) + (3! + 3)] = 5! + 5 \times 29 \quad (13)$$

We observe that $D(n) = (n-1)! + (n-1)g(n-1)$. However, more effort is required to obtain an exact formula for $g(n)$, and thus $D(n)$.

1.1.2 Probability of derangement

Since $N!$ is the number of distinct arrangements of N people/hats, and $D(N)$ is the number of derangements, then $P(N) = \frac{D(N)}{N!}$ is the probability that no-one gets their right hat, assuming that an arbitrary arrangement is selected uniformly at random.

From the recurrence relation, we have

$$P(N) = \frac{D(N)}{N!} \quad (14)$$

$$= \frac{N-1}{N!} [D(N-1) + D(N-2)] \quad (15)$$

$$= \frac{N-1}{N} P(N-1) + \frac{1}{N} P(N-2) \quad (16)$$

$$= P(N-1) - \frac{1}{N} [P(N-1) - P(N-2)] , \quad (17)$$

where $P(0) = \frac{D(0)}{0!} = 1$ by definition, and $P(1) = \frac{D(1)}{1!} = 0$ by construction. The former sort of makes sense when we consider that if there are no people and no hats, then there is no way to correctly assign hats to people, and thus the probability of not correctly assigning hats must be unity!

Unrolling the recursion gives

$$P(2) = P(1) - \frac{1}{2} [P(1) - P(0)] = \frac{1}{2} = \frac{1}{2!} , \quad (18)$$

$$P(3) = P(2) - \frac{1}{3} [P(2) - P(1)] = \frac{1}{2!} - \frac{1}{3} \times \frac{1}{2} = \frac{1}{2!} - \frac{1}{3!} , \quad (19)$$

$$P(4) = P(3) - \frac{1}{4} [P(3) - P(2)] = \left[\frac{1}{2!} - \frac{1}{3!} \right] - \frac{1}{4} \left[\frac{1}{2!} - \frac{1}{3!} \right] + \frac{1}{4} \left[\frac{1}{2!} \right] \quad (20)$$

$$= \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} . \quad (21)$$

We hypothesise that

$$P(N) = \sum_{n=0}^N \frac{(-1)^n}{n!} , \quad (22)$$

which clearly holds for $P(0) = \frac{(-1)^0}{0!} = 1$ and $P(1) = 1 + \frac{(-1)^1}{1!} = 0$. Thus, we have

$$P(N) - P(N-1) = \sum_{n=0}^N \frac{(-1)^n}{n!} - \sum_{n=0}^{N-1} \frac{(-1)^n}{n!} = \frac{(-1)^N}{N!} . \quad (23)$$

Hence, from the recurrence relation, we have

$$P(N+1) = P(N) - \frac{1}{N+1} [P(N) - P(N-1)] \quad (24)$$

$$= \sum_{n=0}^N \frac{(-1)^n}{n!} - \frac{1}{N+1} \frac{(-1)^N}{N!} \quad (25)$$

$$= \sum_{n=0}^N \frac{(-1)^n}{n!} + \frac{(-1)^{N+1}}{(N+1)!} = \sum_{n=0}^{N+1} \frac{(-1)^n}{n!} , \quad (26)$$

and we have therefore proven our hypothesis via induction.

Let us now check this relationship numerically.

```
[1]: def D_n(n):  
    if n <= 0:  
        return 1  
    elif n == 1:  
        return 0  
    else:  
        return (n-1) * (D_n(n-1) + D_n(n-2))
```

```
[2]: print(list(D_n(n) for n in range(21)))
```

```
[1, 0, 1, 2, 9, 44, 265, 1854, 14833, 133496, 1334961, 14684570, 176214841,  
2290792932, 32071101049, 481066515734, 7697064251745, 130850092279664,  
2355301661033953, 44750731559645106, 895014631192902121]
```

```
[3]: def factorial(n):  
    if n <= 1:  
        return 1  
    else:  
        return n * factorial(n-1)
```

```
[4]: def P_n(n):  
    return D_n(n) / factorial(n)
```

```
[5]: print(list(P_n(n) for n in range(21)))
```

```
[1.0, 0.0, 0.5, 0.3333333333333333, 0.375, 0.36666666666666664,  
0.36805555555555556, 0.3678571428571429, 0.36788194444444444,  
0.36787918871252206, 0.3678794642857143, 0.3678794392336059, 0.3678794413212816,  
0.36787944116069116, 0.3678794411721619, 0.3678794411713972, 0.367879441171445,  
0.36787944117144217, 0.36787944117144233, 0.36787944117144233,  
0.36787944117144233]
```

```
[6]: def P_n_sum(n):  
    p = 1      # holds  $(-1)^k / k!$   
    s = 1.0    # holds  $\sum_{k=0}^n (-1)^k / k!$   
    if n >= 1:  
        for k in range(1, n+1):  
            p = -p / k  
            s += p  
    return s
```

```
[7]: print(list(P_n_sum(n) for n in range(21)))
```

```
[1.0, 0.0, 0.5, 0.33333333333333337, 0.37500000000000006, 0.36666666666666667,  
0.36805555555555556, 0.3678571428571429, 0.36788194444444445, 0.3678791887125221,
```

```
0.3678794642857144, 0.367879439233606, 0.3678794413212817, 0.36787944116069127,  
0.36787944117216204, 0.3678794411713973, 0.3678794411714451, 0.3678794411714423,  
0.36787944117144245, 0.36787944117144245, 0.36787944117144245]
```

```
[8]: from math import exp  
  
     print(exp(-1))
```

```
0.36787944117144233
```

We observe that the summation formula for the probability $P(N)$ is just the first N terms in the Maclaurin series expansion of e^{-1} , to which $P(N)$ theoretically converges as $N \rightarrow \infty$. However, we also observe that the summation formula introduces some round-off error that is not present for the counting-based formula, which does correctly converge (numerically) to e^{-1} .