

Paradoxe des anniversaires

Énoncé

- Quelle est la probabilité qu'au moins deux personnes aient leur anniversaire le même jour dans un groupe de 40 personnes ?

Solution

Il est plus simple de passer par le calcul de la probabilité complémentaire $P_{\text{comp}}(N)$, que toutes les N personnes présentes aient leur anniversaire des jours différents. Si on considère une personne à la fois, on multipliera les probabilités indépendantes d'“avoir un anniversaire un jour différent des personnes précédentes” :

- $P_{\text{comp}}(1) = 1 = 365/365$ (trivial pour une seule personne)
- $P_{\text{comp}}(2) = (365/365) * (364/365)$ (pour la deuxième personne, seuls 364 jours sur 365 sont adéquats pour avoir des dates différentes)
- $P_{\text{comp}}(3) = p_{\text{comp}}(2) * (363/365) = 1 * (364/365) * (363/365)$ (pour la troisième personne, seuls 363 jours sur 365 sont adéquats pour avoir des dates différentes)
- ...
- $P_{\text{comp}}(N) = (365 * 364 * 363 * \dots * (365 - N + 1)) / (365)^N = (365)! / ((365 - N)! * (365)^N)$

Pour 40 personnes, la réponse $1 - P_{\text{comp}}(40) = 0.89123$

Remarque : discuter de la non-validité de l'approximation de Stirling utilisée.

Programme Python

[multiples-occurrences-anniversaires.py](#)

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
"""
Petit programme destiné à répondre à la question suivante :
- Quelle est la probabilité qu'au moins deux personnes aient leur
anniversaire
le même jour dans un groupe de 40 personnes.
Hypothèses : l'année fait 365 jours et les naissances des personnes
se distribuent uniformément (pas de jumeaux,...)
Remarque : la solution doit pouvoir être généralisée !
"""

from math import log, exp

# nombre de possibilités différentes
# (= nombre de jours d'une année dans les exemples)
poss = 365

# nombre d'items (personnes présentes)
```

```
n = 40

# solution : p = poss ! / ( (poss-n) ! * poss^n)

pcomp = 1.
for i in range(poss, poss-n, -1):
    pcomp = pcomp * i / poss

print("calcul exact : ", pcomp, 1.-pcomp)

# calcul suivant l'approximation de Stirling ( ln(j!) ~ j ln(j) - j
# l'approximation est d'autant plus valable que poss est grand et n <<
poss
pcomps = exp(poss*log(poss)-poss - (poss-n)*log(poss-n) + (poss-n) -
n*log(poss))

print("Approximation de Stirling : ", pcomps, 1.-pcomps)
```

Problème analogue

- Quelle est la probabilité de recevoir 40 cartes cadeaux différentes (aucun "double") sur 216 types différents de cartes distribuées comme jeu-concours aux caisses d'un supermarché
- Si ce genre d'événement se produit fréquemment, que pouvez-vous en conclure ?
- Discuter des stratégies appliquées par les personnes qui organisent ce genre de jeu-concours

Références

- [W Paradoxe des anniversaires](#)
- [W Birthday problem](#)
- https://www.reddit.com/r/dataisbeautiful/comments/7l9ef7/i_simulated_and_animated_500_instances_of_the/, simulation
- [Using the birthday paradox to teach probability fundamentals - What are the odds that two of your friends share a birthday?](#) Cassie Kozyrkov, Medium, 08/10/2019
- [W Principe des tiroirs](#)
- [W Pigeonhole principle](#)
- Cryptanalyse (informatique)
 - [W Attaque des anniversaires](#)
 - [W Birthday attack](#)
- [Thread by @3blue1brown: "The birthday paradox is very famous in probability. If you take 23 people, there's about a 50/50 chance that two of them share a birthday."](#)

Extrait du thread twitter de @3blue1brown et des commentaires :

The birthday paradox is very famous in probability. If you take 23 people, there's about a 50/50 chance that two of them share a birthday. With 50 people, it's a 97% chance. We could make many other fun examples to illustrate the same counterintuitive phenomenon :

- Choose a random card from a deck of 52 cards. Put it back, shuffle well, and choose another.

- Do this for only 9 draws, and more likely than not, you've pulled the same card twice. Do it 16 times, and your chances are over 90%. Try it!
- Next time you're in an event with more than 118 people, think to yourself that there's a >50% chance that two people there have phone numbers with the same last four digits (assuming those are uniformly distributed). With more than 250 people, it's >95%. Ditto for ATM pin codes.
 - The number of possible poker hands is 2,598,960. One hand for every person in Chicago, or 10 for each mile between the earth and the moon. How many hands would you think you'd have to draw before you've likely had the same *exact* hand twice? It's around 1,900 hands that it becomes more likely than not to have held the same hand twice. That's just 19 evenings with 100 hands each. Many (most?) of you reading this have held the same exact poker hand twice!
 - You can think of the birthday paradox by asking the probability of drawing 23 people successively so that each one has a birthday not yet seen. This gives the probability of no collision, so the probability of a collision is 1 minus this. The general formula for the probability of a collision when making k choices from a collection of N possibilities looks like $1 - \frac{N!/(N-k)!}{N^k}$. By using some approximations for the factorials, it so happens that the inflection point where things shift from collisions being very unlikely to instead being likely is around \sqrt{N} . More specifically, it happens a little above \sqrt{N} . For example, $\sqrt{365} = 19.105\dots$, and it's at 23 people that you're more likely to have a birthday collision than not. In the phone numbers example, $\sqrt{10,000} = 100$, and the 50% point happens with 118 people. In the card example, $\sqrt{52} = 7.2$, and it took 9 draws. Specifically, if you have a hash function with N possible outputs (say 2^{128}), and your system's security depends on collisions never happening, you might initially think an attacker needs around 2^{128} brute force attempts, but really it's much *much* smaller: 2^{64} .

From:

<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:

https://dvillers.umons.ac.be/wiki/teaching:exos:paradoxe_anniversaires

Last update: **2019/11/08 09:56**

