

Experimental birthday paradox

What are the odds that in a group of n randomly chosen people, some of them will have the same birthday? An intuitive guess would suggest that the chance of two individuals sharing the same birthday in a group of 23 is much lower than 50%, but that this is not the case. A 99% probability is even reached with just 57 people. This is called the [birthday paradox](#), because the mathematical truth contradicts naive intuition.

Suppose that a certain type of event has m possible outcomes. For example, you can throw a dice and get the values one through six, which means in this case $m = 6$, or a birthday can be on one of the 365 days of the year (where we do not take leap years into account), which means $m = 365$. Based on the probability theory, it can be proven that when there are n events, the probability that at least two events have the same outcome is equal to $P_2(n,m) = 1 - \frac{m!}{(m-n)! m^n}$ $\text{\textit{waarbij } } m \in \mathbb{N}, 0 \leq n \leq m$ $\text{\textit{]}}$ In this exercise, however, we will not start from this formula, but we will experimentally demonstrate the birthday paradox.

Assignment

1. Write a function `happen_together` that generates n random outcomes for an event type with m possible outcomes, and returns whether at least two events have the same outcome (True) or not (False). The values of m and n should be passed to the function as arguments.

Hint: use a collection to draw events that have already occurred

2. Write a function `estimate_chance` which calculates the probability that at n random events with m possible outcomes, at least two events have the same outcome. Here, the function must successively perform a test a number of times where n random outcomes are generated, and re-occurring outcomes are frequently checked (in which case the function `happen_together` can be useful). The probability is then calculated as the number of tests in which certain outcomes occurred several times divided by the number of tests performed. The values of m , n , and the number performed tests must be passed to the function as a parameter, and the function must return the calculated odds as a result.

Note with evaluation: The functions `happen_together` and `estimate_chance` will be tested by examining whether the estimated probability lies close enough to the exact odds. So it is not inconceivable (but unlikely) that your source code is completely correct and the test still fails. In that case, you can just submit again. This is after all an assignment about probability.

Example

```
>>> happen_together(6, 3)
True
>>> happen_together(6, 3)
False
>>> estimate_chance(6, 2, 10000)
0.1676
>>> estimate_chance(365, 23, 10000)
0.5024
```

Wat is de kans dat er in een groep van n willekeurige personen minstens twee personen dezelfde verjaardag hebben? Je zou het misschien niet verwachten maar voor een groep van

slechts 23 personen is deze kans reeds groter dan 50%. Bij 57 personen is de kans zelfs als meer dan 99%. Dit is het typevoorbeeld van een paradox die men in de kansrekening de *verjaardagsparadox* is gaan noemen en die een resultaat toont dat tegen de intuïtie ingaat.

Stel algemeen dat een bepaald type gebeurtenis m mogelijke uitkomsten heeft. Zo kun je bijvoorbeeld met een dobbelsteen de waarden één tot en met zes gooien waardoor in dit geval $m = 6$, of kan een verjaardag op één van de 365 dagen van het jaar vallen (waarbij we geen rekening houden met schrikkeljaren) waardoor in dit geval $m = 365$. Op basis van kansrekening kan men aantonen dat wanneer er zich n gebeurtenissen voordoen, de kans dat minstens twee gebeurtenissen dezelfde uitkomst hebben gelijk is aan $1 - \frac{m!}{(m-n)! m^n}$ waarbij $m \in \mathbb{N}$, $0 \leq n \leq m$. In deze opgave zullen we niet uitgaan van deze formule, maar zullen we de verjaardagsparadox proefondervindelijk aantonen.

Opgave

1. Schrijf een functie `gebeuren_samen` die voor een type gebeurtenis met m mogelijke uitkomsten, n willekeurige uitkomsten genereert en teruggeeft of er zich minstens twee gebeurtenissen met dezelfde uitkomst voordoen (`True`) of niet (`False`). De waarden voor m en n moeten als argumenten aan de functie doorgegeven worden.

Hint: gebruik een verzameling om gebeurtenissen voor te stellen die zich reeds hebben voorgedaan

2. Schrijf een functie `schat_kans` die de kans berekent dat er bij n willekeurige gebeurtenissen met m mogelijke uitkomsten, minstens twee gebeurtenissen zijn met dezelfde uitkomst. Hierbij moet de functie een aantal keer na elkaar een test uitvoeren waarbij n willekeurige uitkomsten gegenereerd worden en telkens wordt nagegaan of er zich uitkomsten meermaals voordoen (waarbij je dus handig gebruik kan maken van de functie `gebeuren_samen`). De kans wordt dan berekend als het aantal testen waarin bepaalde uitkomsten meermaals voorkwamen gedeeld door het aantal uitgevoerde testen. De waarden voor m , n en het aantal uit te voeren testen moeten als parameter aan de functie doorgegeven worden, en de functie moet de berekende kans als resultaat teruggeven.

Opmerking bij beoordeling: De functies `gebeuren_samen` en `schat_kans` zullen getest worden door na te gaan of de geschatte kans voldoende dicht licht bij de exacte kans. Het is dus niet ondenkbaar (maar wel onwaarschijnlijk) dat je broncode helemaal juist is en de test toch faalt. In dat geval kan je gewoon opnieuw inzenden. Dit is per slot van rekening een opgave over kansrekening.

Voorbeeld

```
>>> gebeuren_samen(6, 3)
True
>>> gebeuren_samen(6, 3)
False
>>> schat_kans(6, 2, 10000)
0.1676
>>> schat_kans(365, 23, 10000)
0.5024
```