LOI DE BENFORD

1. Un premier exemple

Dans ce document, nous allons étudier la loi de Benford (ou loi de Newcomb-Benford), qui est un curieux résultat, à la fois théorique et empirique, lié à notre système de numération. Commençons par un exemple concret.

L'INSEE met à disposition les statistiques de la population française selon plusieurs découpages administratifs : arrondissements, communes, départements, etc. :

`https://catalogue-donnees.insee.fr/fr/catalogue/recherche/DS_POPULATIONS_REFERENCE`

En 2023, la commune de Paris a une population recensée de 2 103 778, celle du 1er arrondissement de Marseille 37 599 et celle de Châteauvieux-les-Fossés de 9 habitants.

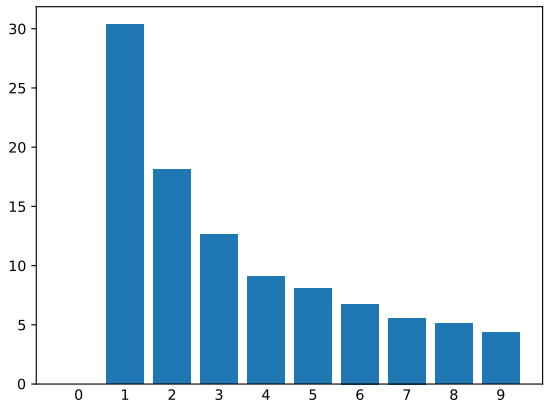
Les ordres de grandeurs sont donc très variées pour les 34 852 communes répertoriées. Nous allons nous poser la question suivante : à quelles fréquences le premier chiffre de ces données est 1, 2, 3, 4, etc. ? Dans les exemples ci-dessus, les premiers chiffres sont respectivement 2, 3 et 9.

À première vue, la question parait étrange. Les données sont tellement variables qu'elles paraissent aléatoires, donc que les probabilités de rencontrer 1, 2, 3, 4,... comme premier chiffre devraient intuitivement être de $\frac{1}{9} \approx 11\,\%$ (loi uniforme). Testons cela.

Benford-population.py

```python
import pandas as pa
import numpy as np
import matplotlib.pyplot as plt

T = pa.read_csv("DS_POPULATIONS_REFERENCE_2023_data.csv", sep=";")
V = T.query("GEO_OBJECT=='COM' and POPREF_MEASURE=='PMUN' and
 ↪  OBS_VALUE>0")['OBS_VALUE']

X = range(10)
Y = np.zeros(10)  # tableau initialisé avec 10 zéros. Y[3] = nb de "3"

for k in V.keys():
   pc = int(str(V[k])[0])   # premier chiffre (pc) extrait en transformant
    ↪  provisoirement le nombre en chaine de caractères
   Y[pc] = Y[pc] + 1        # on ajoute 1 dans la pc-ième case de Y

Y = 100*Y/sum(Y)            # pourcentages
print(Y)

fig, ax = plt.subplots()
ax.bar(X, Y)
plt.xticks(X, labels=X)                # labels des chiffres
plt.tick_params(axis='x', length=0)    # suppression des graduations
plt.show()
```

```
[ 0.          30.34259153
18.13956157 12.61046712
9.07838861   8.08561919
6.7571445    5.57500287
5.09583381   4.3153908 ]
```



Le graphique est étonnant. Le « 1 » apparait environ 30 % du temps, le « 2 » environ 18 % et la fréquence décroit jusqu'au « 9 » (environ 4 %). Pourquoi ?

Cette répartition va en réalité s'observer dans tout jeu de données « naturelles », suffisamment fourni **et** mélangeant différents ordres de grandeurs. Par exemple, pour des superficies de forêts, des masses de corps célestes ou des longueurs de cours d'eau.

Écrivons une donnée quelconque sous forme scientifique $a \times 10^k$ avec $a \in [1\,;10[$ et $k \in \mathbb{Z}$. La partie entière de a, égale à 1, 2, 3, ..., 9 correspond au premier chiffre du nombre. Les éléments d'un jeu de données « grandissent » de manière multiplicative et non additive. Leurs mesures se répartissent uniformément mais selon une progression logarithmique. La probabilité que a appartienne à l'intervalle $[2^0\,;2^1[$ est la même qu'elle appartienne à $[2^1\,;2^2[$ ou à $[2^2\,;2^3[$. Or, les intervalles $[1\,;2[$, $[2\,;4[$ et $[4\,;8[$ n'ont pas la même « longueur ».

S'il y a statistiquement autant de mesures avec $a \in [1\,;2[$ qu'avec $a \in [2\,;4[$, alors il y en aura nécessairement moins dans les intervalles $[2\,;3[$ et $[3\,;4[$ pris séparément. Cela est similaire pour les intervalles $[4\,;5[$, $[5\,;6[$,... qui, chacun, contiennent de moins en moins de mesures. Cette décroissance se calcule : c'est l'objet de la prochaine partie.

2. Expression mathématique

Soit X une variable aléatoire représentant un nombre dans un jeu de données « naturelles ».

On note $X = a \times 10^k$ et on a $\log(X) = \log(a) + k$

On note $Y = \log(a)$. L'hypothèse est que la v.a.r. Y suit une loi uniforme dans $[0\,;1[$

Soit c l'entier entre 1 et 9 tel que $\log(c) \leqslant Y < \log(c+1)$

Puisque Y est uniformément distribuée dans $[0\,;1[$, la probabilité qu'elle tombe dans l'intervalle $[\log(c)\,;\log(c+1)[\subset [0\,;1[$ est égale à $P(c) = \log(c+1) - \log(c) = \log\left(\frac{c+1}{c}\right)$

Les valeurs de `[round(log((c+1)/c, 10), 4) for c in range(1, 10)]` nous donnent le tableau :

1	2	3	4	4	6	7	8	9
$\log\left(\frac{2}{1}\right)$	$\log\left(\frac{3}{2}\right)$	$\log\left(\frac{4}{3}\right)$	$\log\left(\frac{5}{4}\right)$	$\log\left(\frac{6}{5}\right)$	$\log\left(\frac{7}{6}\right)$	$\log\left(\frac{8}{7}\right)$	$\log\left(\frac{9}{8}\right)$	$\log\left(\frac{10}{9}\right)$
0,3010	0,1761	0,1249	0,0969	0,0792	0,0669	0,058	0,0512	0,0458

3. Benford or not Benford ?

À la fin du XIX^{e} siècle, l'astronome américain Simon Newcomb remarque les premières pages des tables de logarithmes sont plus usées que les suivantes. Il publie une note sur ce phénomène et propose une théorie qui sera redécouverte en 1938 par l'ingénieur Frank Benford. Ce dernier analyse plus de 20 000 observations provenant de sources scientifiques et journalistiques, et les statistiques confirment cette répartition logarithmique des premiers chiffres.

De nombreuses données humaines suivent de plus ou moins près la loi de Newcomb-Benford : prix des produis dans les magasins, résultats sportifs, cours d'une action en bourse, etc. Le changement d'unités ou la conversion dans une autre devise n'a pas d'influence. Il est nécessaire que les données soient suffisamment nombreuses et étalées[1].

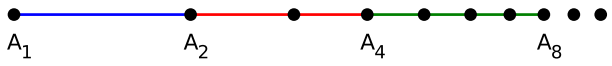
Mathématiquement, certaines suites de nombres suivent la loi de Benford : Fibonacci, $(n!)_n, (n^n)_n$, quelque soit la base. Les puissances d'un entier m la suivent dans n'importe quelle base b telle que $\log_b(m) \notin \mathbb{Q}$.

D'autres données sont trop artificielles pour suivre cette loi : numéros de téléphone, codes postaux, dates, adresses IP, etc. Si l'on génère des nombres aléatoires avec un ordinateur, les premiers chiffres sont présents uniformément. Si l'on demande à des humains de donner des nombres au hasard, les premiers chiffres ont une fréquence biaisée.

Ces constatations ont amené différentes entités à utiliser la loi de Benford pour détecter des fraudes : comptabilités douteuses, données scientifiques contrefaites, manipulations électorales, etc.

4. À vous de jouer !

1. Tester la loi de Benford avec : (a) la suite de Fibonacci ; (b) la suite $(2^n)_n$; (c) un autre jeu de données publiques.
2. Quelle ligne changer dans le code proposé pour étudier la répartition du **dernier** chiffre ? Que donne l'histogramme ?
3. Générer un lot de 10 000 entiers respectant « suffisamment bien » la loi de Benford.
4. Tracer 10 points $(A_c)_c$ répartis tels que $\forall c \in [\![1, 9]\!]$, $d(A_c, A_{c+1}) = \log\left(\frac{c+1}{c}\right)$ (voir ci-dessous)
5. Pourquoi a-t-on $\sum_{c=1}^{9} \log\left(\frac{c+1}{c}\right) = 1$?



5. Références

- Nicolas Gauvrit et Jean-Paul Delahaye, *Pourquoi la loi de Benford n'est pas mystérieuse*, Mathématiques et sciences humaines, 182 – 2008, 7-15. `https://journals.openedition.org/msh/10363`
- Jean-Paul Delahaye, *Une explication pour la loi de Benford*, Pour la science n°489, juillet 2018
- A. Berger, T. P. Hill, and E. Rogers, *Benford Online Bibliography*, 2009 (consulté en 2026) `https://benfordonline.net`

1. Si on ne s'intéressait qu'au nombre de buts par match au football, par exemple, les chiffres supérieurs à 5 seraient quasi inexistants.