

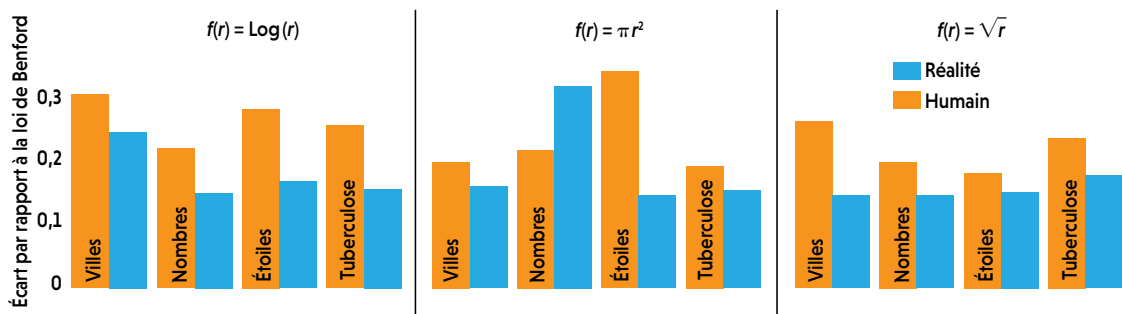
LA LOI DE BENFORD GÉNÉRALE SOUMISE À DES TESTS

4

La loi de Benford générale affirme que la partie fractionnaire des nombres $f(r)$ obtenus à partir d'une série de nombres r sera uniformément étalée sur l'intervalle $[0, 1]$ et cela pour toute fonction continue croissante f . Comme la loi de Benford classique, fondée sur $f(r) = \log_{10}(r)$, on peut utiliser la loi générale pour repérer des données truquées.

Afin de s'assurer que la méthode est bonne, on a demandé à des sujets humains de créer de fausses données et on a mesuré la qualité de ce qu'ils produisaient en utilisant la loi de Benford générale pour trois fonctions différentes : $f(r) = \log(r)$, $f(r) = \pi r^2$ et $f(r) = \sqrt{r}$. On a utilisé des données provenant de quatre sources différentes qui ont conduit à 12 comparaisons

entre la réalité et ce que produit un humain simulant la réalité (voir le protocole détaillé dans le texte principal). Dans 11 cas sur 12, la simulation par l'humain est moins conforme à la loi de Benford que les données réelles. Il est donc possible, en effectuant des tests fondés sur la loi de Benford, de détecter des données artificielles produites par des humains.



> proportion est $\log_{10}(c+1) - \log_{10}(c)$. C'est exactement ce qu'exprime la loi de Benford formulée au sujet du premier chiffre significatif en base décimale!

La loi de Benford continue est effectivement un peu plus puissante que la loi n'évoquant que les chiffres significatifs en base 10. Elle permet par exemple de retrouver, dans toute base de numération, un énoncé évoquant le premier chiffre significatif.

La justification de l'équivalence utilisée plus haut se fait rigoureusement, mais un exemple clarifiera l'équivalence. Prenons $r=7234$. On a :

$$\begin{aligned} \{\log_{10}(7234)\} &= \log_{10}(7234) - [\log_{10}(7234)] \\ &= \log_{10}(7234) - [3,8593...] = \log_{10}(7234) - 3 \\ &= \log_{10}(7234) - \log_{10}(10^3) = \log_{10}(7234/1000) \\ &= \log_{10}(7,234). \end{aligned}$$

Puisque $\log_{10}$ est une fonction croissante, on a : $\log_{10}(7) < \log_{10}(7,234) < \log_{10}(8)$.

Soustraire $[\log_{10}(r)]$ à $\log_{10}(r)$ ramène la valeur dans le $\log_{10}$ entre 1 et 10 et donc l'encadrement entre $\log_{10}(c)$ et $\log_{10}(c+1)$ repère le premier chiffre significatif. D'où l'équivalence indiquée.

ON VEUT DES PREUVES!

Comme précédemment, le principe énoncé, c'est-à-dire la loi de Benford continue, est un peu vague, et il faudrait préciser quand elle ne s'applique pas, à l'aide de résultats mathématiques démontrés.

De tels énoncés mathématiques existent; ils sont un peu compliqués à formuler (voir l'article de Michel Valadier cité dans la

bibliographie). Ils consistent à formuler des hypothèses exprimant l'idée d'étalement et de régularité, et, en fonction de la précision avec laquelle les hypothèses sont imposées, ils affirment que la série de nombres vérifie la loi de Benford avec une précision que le théorème explicite.

Il faut être un peu prudent avec la formulation non formelle de la loi de Benford; elle n'est que la traduction d'une intuition « invincible » comme l'écrit Poincaré, mais qui reste une intuition. Tout dépend de ce qu'on nomme « large plage de plusieurs ordres de grandeur » et « loi étalée et régulière ». Cependant, l'énoncé informel montre pourquoi la loi est souvent vérifiée, au moins approximativement.

La loi informelle explique aussi une autre propriété de la loi de Benford remarquée depuis longtemps: en augmentant la taille d'une série de nombres, on ne tend pas toujours vers les valeurs annoncées des probabilités $\log_{10}(c+1) - \log_{10}(c)$. L'explication est claire: si les nombres suivent une loi précise, la compensation entre les intervalles quand on passe aux parties fractionnaires des logarithmes ne se fera qu'exceptionnellement, et donc en multipliant les données, on ne converge pas vers une loi parfaitement uniforme sur $[0, 1]$, mais vers une loi approximativement uniforme.

Comprendre est le meilleur moyen d'avancer et c'est ici le cas. L'identification de l'origine de la loi de Benford suggère une méthode simple de la généraliser: remplacer la fonction