L'ENTROPIE EN ACTION

13 activités à partir de données réelles — Terminale

Niveau : Terminale générale, Maths complémentaires. Approfondissement.

Lien avec le programme : Thème « **Fonction logarithme népérien** » (changement de base), « **Variables aléatoires : espérance** ». Ce thème peut s'étendre à la loi binomiale, aux probabilités conditionnelles, à l'étude de fonctions et à la programmation en Python.

Lien avec Les maths au quotidien : Codage/IA/Société/Santé/Loisirs/Représentation visuelle/Modèles économiques

Méthode : calculer l'entropie

Le programme de Terminale ne connaît que le logarithme népérien ln (touche « ln » de la calculatrice). On calcule le logarithme en base 2 avec la formule de changement de base :

$$\log_2(x) = \frac{\ln(x)}{\ln(2)}$$

Pour une source qui produit n résultats de probabilités $p_1, \dots, p_n$ (avec $p_1 + \cdots + p_n = 1$), l'**entropie de Shannon** est l'espérance de la quantité d'information $-\log_2 p_i$:

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i = \sum_{i=1}^{n} p_i \log_2\left(\frac{1}{p_i}\right) \text{ bits}$$

Elle vaut 0 si un résultat est certain, et son maximum $\log_2 n$ si les n résultats sont équiprobables. Dans toutes les activités, **les données sont réelles** (mesures, statistiques officielles, constantes techniques) : la source est indiquée sous chaque activité.

N°	Activité	Maths au quotidien
1	Fichiers : l'entropie d'un octet	Codage/Société
2	Réseaux : détecter un scan de ports	Codage/Société
3	Inégalités de niveau de vie (Insee)	Société/Modèles économiques
4	ADN : l'entropie d'un génome	Santé/Codage
5	Aléa : la faille Debian/OpenSSL	Codage/Société
6	Imagerie : mesurer la netteté	Représentation visuelle/Santé/Astronomie
7	Jeux : 20 questions et Wordle	Loisirs/Codage
8	Langues : la redondance du texte	Codage/Société
9	Compression : l'arbre de Huffman	Codage/Représentation visuelle
10	Télécoms : le débit d'un câble	Codage/Bricolage
11	Mots de passe : combien de temps ?	Codage/Société
12	Intelligence artificielle : le Titanic	IA/Société/Navigation
13	Corriger les erreurs : le canal bruité	Codage/Astronomie

Activité 1 — Fichiers : l'entropie d'un octet

Lien avec Les maths au quotidien : Codage/Société

Un fichier est une suite d'**octets**, chacun pouvant prendre 256 valeurs. En notant p_i la fréquence de la valeur i dans le fichier, on calcule son entropie H en bits par octet. Les logiciels de sécurité s'en servent pour repérer les fichiers chiffrés ou « packés » (dissimulés) : leur contenu est quasi aléatoire. Voici des mesures réalisées avec ce programme Python :

```
from collections import Counter
from math import log2
def entropie(octets):
    n = len(octets)
    return -sum(c/n*log2(c/n) for c in Counter(octets).values())
```

Fichier	Taille (octets)	Entropie mesurée (bit/octet)
Texte de la licence GPL v3 (anglais)	35 149	4,573
Programme Linux */bin/ls*	142 312	5,935
Photo au format JPEG	61 306	7,937
Le texte GPL compressé avec gzip	12 148	7,985
Le texte GPL chiffré (AES-256)	35 168	7,995
35 149 octets tirés au hasard	35 149	7,994

1. Combien de valeurs un octet peut-il prendre ? Calculer l'entropie maximale d'un octet avec la formule de changement de base.
2. Un code qui traite chaque octet isolément ne peut pas descendre, en moyenne, sous H bits par octet. Quelle taille minimale (en octets) peut avoir le texte GPL ainsi compressé ?
3. Or gzip a obtenu 12 148 octets. Y a-t-il une contradiction avec la question 2 ? Expliquer.
4. On compresse à nouveau le fichier gzip avec gzip : on obtient 12 171 octets. Interpréter à l'aide du tableau.
5. Un logiciel signale comme suspect tout fichier d'entropie supérieure à 7,5 bit/octet. Quels fichiers du tableau sont signalés ? Lequel est un « faux positif » ?

Sources : mesures réalisées avec le programme ci-dessus ; Lyda et Hamrock, « Using Entropy Analysis to Find Encrypted and Packed Malware », *IEEE Security & Privacy*, 2007.

Activité 2 — Réseaux : détecter un scan de ports

Lien avec Les maths au quotidien : Codage/Société

Un ordinateur reçoit des connexions sur des **ports**, numérotés de 0 à 65 535 (un numéro est codé sur 16 bits). Un outil de surveillance calcule régulièrement l'entropie H de la répartition des ports visés. Une valeur très éloignée de l'habitude du réseau est le signe d'une anomalie (Lakhina, Crovella et Diot, 2005). Deux attaques classiques :

- une **saturation** (SYN flood) : des millions de connexions visent le même port, par exemple le 443 d'un site web ;
- un **scan de ports** : un pirate cherche les portes ouvertes ; par défaut, le logiciel Nmap teste les 1 000 ports les plus courants, à fréquence égale.

1. Combien de ports existe-t-il ? Calculer l'entropie maximale de la répartition des ports visés.
2. Calculer l'entropie H pendant une saturation visant un seul port.
3. Calculer l'entropie H pendant un scan Nmap par défaut.
4. Un scan complet teste les 65 536 ports à fréquence égale. De combien de bits l'entropie est-elle supérieure à celle du scan par défaut ? Par quel facteur le nombre de ports testés est-il multiplié ?
5. L'entropie habituelle d'un réseau fluctue autour d'une valeur H_0. Pourquoi une **baisse** brutale et une **hausse** brutale de H peuvent-elles toutes deux signaler une attaque ?

Sources : documentation de Nmap (1 000 ports les plus courants par défaut) ; A. Lakhina, M. Crovella, C. Diot, « Mining Anomalies Using Traffic Feature Distributions », ACM SIGCOMM, 2005.

Activité 3 — Inégalités de niveau de vie (Insee)

Lien avec Les maths au quotidien : Société/Modèles économiques

On classe la population de la France métropolitaine du plus modeste au plus aisé et on la découpe en **10 groupes de même taille** (déciles). L'Insee publie la part de la masse totale des niveaux de vie détenue par chaque groupe (année 2024, en %). Les parts sont arrondies : leur somme fait 100,1.

Groupe	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
Part (%)	3,2	5,1	6,3	7,3	8,2	9,2	10,3	11,7	13,9	24,9

Si les revenus étaient parfaitement égaux, chaque groupe détiendrait la même part. L'**écart** $I = \log_2 n - H$ entre l'entropie maximale et l'entropie réelle de la répartition mesure donc l'inégalité (en économie, $I \times \ln 2$ est l'indice de Theil).

1. Regrouper les déciles deux par deux pour obtenir les parts des 5 **quintiles**, puis les diviser par 100,1 pour en faire des probabilités p_i.
2. Calculer l'entropie H de cette répartition en 5 groupes.
3. Calculer $\log_2 5$, puis l'écart $I = \log_2 5 - H$.
4. Que vaudrait I si un seul groupe détenait tout le revenu ? Si le revenu était parfaitement égal ? Interpréter.
5. À l'aide d'un tableur, refaire le calcul avec les 10 déciles. Pourquoi l'écart I obtenu est-il plus grand ?

Source : Insee, enquête Revenus fiscaux et sociaux 2024 (Insee-DGFiP-Cnaf-Cnav-CCMSA), part de la masse totale de niveau de vie détenue selon la tranche de niveau de vie, France métropolitaine.

Activité 4 — ADN : l'entropie d'un génome

Lien avec Les maths au quotidien : Santé/Codage

L'ADN s'écrit avec 4 bases : A, T, C, G. La règle de Chargaff dit que A et T sont aussi fréquents l'un que l'autre, de même que C et G. Si g désigne le **taux de GC** (la proportion de bases G ou C), on a donc $p_A = p_T = \frac{1-g}{2}$ et $p_C = p_G = \frac{g}{2}$. Le taux de GC varie beaucoup selon les espèces :

Organisme	Taux de GC g
Parasite du paludisme (*Plasmodium falciparum*)	≈ 0,20
Être humain	0,41
Bactérie *Streptomyces coelicolor*	0,72

1. Quelle est l'entropie maximale d'une base ? À quelle situation correspond-elle ?
2. Pour l'être humain ($g = 0{,}41$), calculer p_A, p_C, puis l'entropie H d'une base.
3. Montrer que, pour tout g : $H = 1 + h(g)$ avec $h(g) = -g\log_2 g - (1-g)\log_2(1-g)$.
4. Calculer H pour les deux autres organismes. Les classer. Quel lien avec l'équilibre des bases ?
5. Le génome humain compte environ $3{,}1 \times 10^9$ bases. Quelle taille (en Mo) à 2 bits par base ? Quelle taille minimale avec un codage base par base de longueur moyenne H ?
6. Étudier les variations de h sur $]0\,;1[$ (calculer h') et en déduire pour quelle valeur de g l'entropie de l'ADN est maximale.

Sources : taux de GC des génomes (Wikipédia, « GC-content », d'après l'International Human Genome Sequencing Consortium 2001 et les données NCBI) ; taille du génome humain : environ 3,1 milliards de bases.

Activité 5 — Aléa : la faille Debian/OpenSSL

Lien avec Les maths au quotidien : Codage/Société

Une clé de chiffrement n'est sûre que si elle est tirée au hasard parmi un très grand nombre de valeurs. Du 17 septembre 2006 au 13 mai 2008, une modification du paquet OpenSSL de Debian (système Linux très répandu) a supprimé presque tout l'aléa du générateur de clés : le seul élément variable restant était le numéro de processus (PID) du programme, qui ne dépassait pas **32 768**. Des milliers de clés SSH et SSL, censées être imprévisibles, ne pouvaient en fait prendre que 32 768 valeurs (faille CVE-2008-0166).

1. En supposant les 32 768 valeurs équiprobables, calculer l'entropie réelle de ces clés.
2. Une clé aurait dû avoir 128 bits d'entropie (comme une clé AES-128). Combien de bits d'entropie ont été perdus ? Quelle fraction de l'espace de clés possibles est réellement utilisée ?
3. Un attaquant teste 1 clé par seconde. Combien de temps lui faut-il pour essayer toutes les clés faibles ? Et pour essayer les 2^{128} clés d'un générateur correct (en années) ? Comparer à l'âge de l'Univers, environ $1{,}4 \times 10^{10}$ ans.
4. Avec 10^6 essais par seconde, combien de temps pour les clés faibles ?
5. Expliquer pourquoi la longueur apparente d'une clé ne suffit pas à garantir sa sécurité.

Sources : Debian Security Advisory DSA-1571 (13 mai 2008) ; R. Finnie, « I discovered the Debian OpenSSL bug » (2024) ; entrée CVE-2008-0166.

Activité 6 — Imagerie : mesurer la netteté d'une image

Lien avec Les maths au quotidien : Représentation visuelle/Santé/Astronomie

En IRM comme en astronomie, le mouvement ou une mauvaise mise au point **étale** le signal sur plus de pixels. Pour corriger automatiquement le flou, des chercheurs ont proposé de choisir l'image d'**entropie minimale** : l'entropie de la répartition spatiale de l'intensité, où p_i est la part de l'intensité totale portée par le pixel i (critère d'entropie de Atkinson et al., 1997).

Une étoile est photographiée : son image nette tient sur **un seul pixel**. Le flou le plus courant en traitement d'image est le flou « binomial », qui répartit l'intensité d'un pixel sur les 9 pixels d'un carré 3 × 3 avec les poids (calculés avec le triangle de Pascal 1-2-1) :

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

1. Vérifier que la somme des 9 poids vaut 1. Quelle est l'entropie de l'image nette ?
2. Calculer l'entropie H_3 de l'image floue 3 × 3.
3. Quelle est l'entropie maximale pour 9 pixels ? Pourquoi H_3 ne l'atteint-elle pas ?
4. Un flou plus fort utilise le carré 5 × 5 de poids $\frac{a_i a_j}{256}$ avec $(a_1, \ldots, a_5) = (1{,}4{,}6{,}4{,}1)$. Comme les poids sont des produits :

$$H_5 = 2 \times H\left(\frac{1}{16}, \frac{4}{16}, \frac{6}{16}, \frac{4}{16}, \frac{1}{16}\right)$$

   Calculer H_5.
5. Ranger les trois images selon l'entropie. Pourquoi un algorithme d'autofocus a-t-il intérêt à **minimiser** l'entropie ?

Sources : D. Atkinson et al., « Automatic correction of motion artifacts in magnetic resonance images using an entropy focus criterion », *IEEE Trans. Medical Imaging*, 1997 ; flou gaussien approché par le noyau binomial 1-2-1.

Activité 7 — Jeux : 20 questions et Wordle

Lien avec Les maths au quotidien : Loisirs/Codage

Le jeu des 20 questions. Un joueur pense à un objet, un animal ou un personnage (le « secret »). L'autre doit le deviner en posant au plus 20 questions, auxquelles on ne peut répondre que par **oui** ou par **non** (« Est-ce un animal ? », « Est-il plus grand qu'un chat ? »). Une bonne question partage les possibilités en deux groupes à peu près égaux.

Wordle. Dans ce jeu en ligne (version originale en anglais), un mot secret de 5 lettres est tiré au hasard. À chaque essai (6 au maximum), le joueur propose un mot existant de 5 lettres et le jeu colore chaque lettre : **vert** si elle est dans le mot secret à cette place, **jaune** si elle est dans le mot secret mais à une autre place, **gris** si elle n'y est pas. Exemple avec le mot secret CRANE et l'essai TRACE :

Lettre essayée	T	R	A	C	E
Couleur obtenue	gris	vert	vert	jaune	vert

T n'est pas dans CRANE ; R, A et E sont à la bonne place ; C est dans CRANE, mais en première position. Chaque essai renvoie ainsi un « motif » de 5 couleurs, dont le joueur déduit des informations sur le mot secret. La liste d'origine du jeu contient **2 315** mots-solutions (et 12 972 mots acceptés en tout).

1. Avec n questions oui/non, combien d'objets peut-on distinguer au maximum ? Combien faut-il de questions pour identifier un objet parmi un million ?
2. On cherche un nombre entier entre 1 et 100. Combien de questions faut-il au minimum, avec la bonne stratégie ? Laquelle ?
3. Une question dont la réponse « oui » a la probabilité p apporte en moyenne :
$$h(p) = -p\log_2 p - (1-p)\log_2(1-p) \text{ bit}$$
Comparer « Le nombre est-il supérieur à 50 ? » ($p = \frac{1}{2}$) et « Le nombre est-il 37 ? » ($p = \frac{1}{100}$).
4. Quelle quantité d'information faut-il pour désigner un des 2 315 mots de Wordle ?
5. Un essai renvoie 5 cases de 3 couleurs possibles. Combien de motifs au maximum ? Quelle information maximale un essai peut-il apporter ? En déduire le nombre minimal d'essais nécessaires.

Sources : liste d'origine de Wordle (2 315 solutions, 12 972 mots acceptés) ; analyse de Wordle par la théorie de l'information, chaîne 3Blue1Brown.

Activité 8 — Langues : la redondance du texte

Lien avec Les maths au quotidien : Codage/Société

Fréquences des 26 lettres dans un corpus de textes français (en %, lettres accentuées regroupées avec leur lettre de base ; somme : 99,87 %) :

A	B	C	D	E	F	G	H	I	J	K	L	M
8,70	0,93	3,15	3,55	17,83	0,96	0,97	1,08	6,97	0,71	0,16	5,68	3,23
N	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
6,42	5,35	3,03	0,89	6,43	7,91	7,11	6,12	1,83	0,04	0,42	0,19	0,21

1. Calculer l'entropie maximale d'un texte de 26 lettres, puis de 27 symboles (26 lettres et l'espace).
2. À l'aide d'un tableur (diviser d'abord les fréquences par 99,87 %), calculer l'entropie H_0 des lettres du français. De combien de pour cent est-elle inférieure à $\log_2 26$?
3. En 1951, Shannon a fait deviner à des lecteurs la lettre suivante d'un texte anglais (27 symboles) : il en a déduit que l'entropie réelle est comprise entre 0,6 et 1,3 bit par symbole. Calculer la **redondance** $R = 1 - \frac{H}{\log_2 27}$ pour chacune des deux bornes.
4. À l'activité 1, gzip a compressé un texte de 35 149 caractères en 12 148 octets. Calculer le nombre de bits par caractère obtenu et le situer par rapport aux valeurs précédentes.
5. Pourquoi la redondance d'une langue permet-elle de deviner un mot tronqué ou de corriger une faute de frappe ?

Sources : fréquences des lettres, sttmedia.com (français) ; C. E. Shannon, « Prediction and Entropy of Printed English », *Bell System Technical Journal*, 1951 ; mesures de l'activité 1.

Activité 9 — Compression : l'arbre de Huffman

Lien avec Les maths au quotidien : Codage/Représentation visuelle

Les formats ZIP, PNG, JPEG et MP3 utilisent le codage de Huffman (voir le cours, § 6.1). On l'applique au mot **STATISTIQUES** (12 lettres).

Lettre	S	T	I	A	Q	U	E
Poids (apparitions)	3	3	2	1	1	1	1
Probabilité p_i	$\frac{3}{12}$	$\frac{3}{12}$	$\frac{2}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$

1. Calculer l'entropie H du mot (en bits par lettre).
2. Construire l'arbre de Huffman en complétant, étape par étape, la liste des poids restants (en cas d'égalité, choisir au hasard ; on étiquette 0 la branche vers le plus léger).

Étape	Poids présents à la fin de l'étape	Fusion effectuée
0	S 3 ; T 3 ; I 2 ; A 1 ; Q 1 ; U 1 ; E 1	—
1		
2		
3		
4		
5		
6		

3. Donner le code de chaque lettre et vérifier qu'aucun code n'est le début d'un autre.
4. Calculer la longueur moyenne L des codes. Vérifier que $H \leq L < H + 1$ et le contrôle « somme des poids des nœuds internes ».
5. Combien de bits pour écrire STATISTIQUES avec ce code ? Avec un code de longueur fixe (3 bits par lettre) ? En ASCII (8 bits par lettre) ? Calculer les gains.

Activité 10 — Télécoms : le débit d'un câble

Lien avec Les maths au quotidien : Codage/Bricolage

Le débit maximal d'un canal de bande passante B (Hz) et de rapport signal sur bruit $\frac{S}{N}$ (sans unité) est donné par la formule de Shannon-Hartley. Le rapport signal/bruit est souvent donné en décibels (dB) et se convertit par $\frac{S}{N} = 10^{\mathrm{SNR_{dB}}/10}$.

$$C = B \log_2\left(1 + \frac{S}{N}\right)$$

Technologie	Bande passante B	SNR	Débit annoncé
Modem sur ligne téléphonique (V.34)	3,1 kHz	35 dB	33,6 kbit/s
ADSL2+ (ligne en cuivre)	2,2 MHz	30 dB	24 Mbit/s au maximum
Wi-Fi 802.11n (1 antenne, canal 20 MHz)	20 MHz	30 dB	72,2 Mbit/s au maximum

1. Convertir les rapports signal/bruit de 35 dB et 30 dB en valeurs sans unité.
2. Calculer la capacité de Shannon de chacune des trois technologies.
3. Comparer le résultat du modem au débit annoncé.
4. Pour l'ADSL2+, le débit annoncé dépasse votre résultat. Calculer le rapport signal/bruit minimal (sans unité, puis en dB) pour atteindre 24 Mbit/s sur 2,2 MHz. Sur quelles lignes est-ce possible ?
5. Montrer que doubler la bande passante double la capacité, alors que doubler le rapport $\frac{S}{N}$ (de 1 000 à 2 000) n'ajoute qu'environ B bit/s. Pourquoi le Wi-Fi moderne élargit-il ses canaux ?

Sources : normes ITU-T V.34 et G.992.5 (ADSL2+, bande 2,2 MHz, 24 Mbit/s) ; IEEE 802.11n (MCS 7, 72,2 Mbit/s avec intervalle de garde court) ; bande téléphonique 300-3 400 Hz.

Activité 11 — Mots de passe : combien de temps ?

Lien avec Les maths au quotidien : Codage/Société

L'entropie d'un mot de passe de L caractères tirés **au hasard** parmi N est $H = L\log_2 N$ (il y a N^L possibilités). Vitesses réelles de cassage mesurées en 2022 sur une carte graphique NVIDIA RTX 4090 avec le logiciel hashcat : environ 3×10^{11} essais par seconde pour des empreintes rapides (NTLM) et seulement 2×10^5 par seconde pour des empreintes lentes (bcrypt, coût 5).

Mot de passe	Longueur L	Alphabet (taille N)
① minuscules	8	a-z (26)
② minuscules (phrase de passe)	16	a-z (26)
③ minuscules, majuscules, chiffres, symboles	8	94 caractères imprimables

1. Calculer l'entropie des trois mots de passe, puis leur nombre de combinaisons.
2. Combien de temps faut-il pour les tester toutes avec des empreintes NTLM (3×10^{11} essais/s) ?
3. Même question avec des empreintes bcrypt (2×10^5 essais/s) pour ① et ③ (en jours, puis en années).
4. Comparer ② et ③ pour NTLM. Que confirme-t-on sur les recommandations du NIST (privilégier la longueur) ?
5. Dans la fuite de données RockYou (2009), « 123456 » était le mot de passe de 290 731 des 32,6 millions de comptes. Calculer sa probabilité, puis sa quantité d'information en bits. Comparer à l'entropie du mot de passe ①.

Sources : benchmarks hashcat sur RTX 4090 (forum hashcat, 2022-2023) ; NIST SP 800-63B ; décomptes de la fuite RockYou publiés par SC Magazine (2010).

Activité 12 — Intelligence artificielle : le Titanic

Lien avec Les maths au quotidien : IA/Société/Navigation

Un **arbre de décision** choisit comme première question celle qui apporte le plus fort **gain d'information** : $G = H_{\text{total}} - H_{\text{après}}$, où $H_{\text{après}}$ est la moyenne, pondérée par les effectifs, des entropies des groupes obtenus. On l'applique aux données réelles du naufrage du Titanic (1912) : sur 2 201 personnes à bord, **711 ont survécu**.

Variable « sexe »	Effectif	Survivants	Taux de survie
Femmes	470	344	73,2 %
Hommes	1 731	367	21,2 %

Variable « classe »	Effectif	Survivants	Taux de survie
1re classe	325	203	62,5 %
Autres (2e, 3e, équipage)	1 876	508	27,1 %

1. Calculer l'entropie H_{total} de la survie (probabilité de survie $\frac{711}{2201}$).
2. Calculer les entropies H_{F} et H_{H} des femmes et des hommes.
3. Calculer H_{sexe} (moyenne pondérée) puis le gain G_{sexe}.
4. Faire de même avec la variable « classe » pour obtenir G_{classe}.
5. Quelle variable l'arbre interroge-t-il en premier ? Est-ce cohérent avec la consigne « les femmes et les enfants d'abord » ?

Source : jeu de données du naufrage du Titanic (R. J. M. Dawson, 1995), 2 201 personnes, passagers et équipage.

Activité 13 — Corriger les erreurs : le canal bruité

Lien avec Les maths au quotidien : Codage/Astronomie

Les sondes spatiales, les CD ou les QR codes subissent du bruit qui inverse certains bits. Modèle du canal binaire symétrique : chaque bit est inversé avec la probabilité $p = 0{,}1$, indépendamment des autres. Pour se protéger, on envoie **chaque bit 3 fois** et le récepteur retient la valeur majoritaire (code à répétition).

1. Le nombre X d'erreurs parmi les 3 copies suit la loi binomiale $\mathrm{B}(3\,;0{,}1)$. Quelle est la probabilité que le bit décodé soit faux ?
2. Même question avec 5 copies (la majorité se décide à 3 sur 5).
3. Quel est le rendement (nombre de bits utiles par bit envoyé) de chaque code ?
4. Un message de 1 000 bits est envoyé sans protection. Quelle est l'espérance du nombre de bits faux ? Et avec le code à répétition par 3 ?
5. Shannon montre que le canal permet un rendement aussi proche qu'on veut de $C = 1 - h(p)$ avec une erreur aussi petite qu'on veut. Calculer C pour $p = 0{,}1$ et comparer au rendement du code à répétition.
6. Pour quelle valeur de p la capacité est-elle nulle ? Interpréter.

Sources : C. E. Shannon, « A Mathematical Theory of Communication », 1948 (théorème du codage de canal) ; niveaux de correction d'erreur des QR codes : 7 %, 15 %, 25 % et 30 % (norme ISO/IEC 18004).

Corrigés détaillés

Corrigé de l'activité 1 — Fichiers

1. Un octet compte 8 bits, donc $2^8 = 256$ valeurs. Entropie maximale :

$$\log_2 256 = \frac{\ln 256}{\ln 2} = \frac{8\ln 2}{\ln 2} = 8 \text{ bits par octet}$$

2. Taille minimale : $35\,149 \times \frac{4{,}573}{8} \approx 20\,093$ octets, soit environ 57 % de la taille d'origine.

3. Pas de contradiction : la borne vaut pour un code qui traite **chaque octet isolément**. gzip repère aussi les **répétitions** (mots, motifs qui reviennent) : il exploite des liens entre les octets, ce que l'entropie « octet par octet » ne mesure pas. Il obtient $\frac{12\,148}{35\,149} \approx 35$ % de la taille d'origine.

4. Le fichier gzip a une entropie de 7,985 bit/octet, proche du maximum 8 : ses octets sont presque équiprobables, il n'y a plus de redondance à exploiter. Recompresser ne fait que **ajouter** l'en-tête de gzip : 12 171 octets, un peu plus que 12 148.

5. Sont signalés : le fichier gzip (7,985), la photo JPEG (7,937), le fichier chiffré (7,995) et les octets aléatoires (7,994). Le texte (4,573) et le programme (5,935) ne le sont pas. La photo JPEG est un **faux positif** : elle est légitime mais déjà compressée, donc son entropie est élevée. En pratique, l'entropie est un indice parmi d'autres.

Corrigé de l'activité 2 — Réseaux

1. $65\,536 = 2^{16}$ ports. Entropie maximale : $\log_2 65\,536 = 16$ bits.

2. Un seul port visé avec la probabilité 1 : $H = 1 \times \log_2 1 = 0$ bit.

3. 1 000 ports équiprobables :

$$H = \log_2 1\,000 = \frac{\ln 1000}{\ln 2} \approx 9{,}97 \text{ bits}$$

4. $16 - 9{,}97 \approx 6{,}03$ bits de plus. Le nombre de ports est multiplié par :

$$\frac{65\,536}{1\,000} \approx 65{,}5 \approx 2^{6{,}03}$$

Chaque bit d'entropie supplémentaire double le nombre de ports.

5. Une saturation concentre le trafic sur un port : H chute vers 0. Un scan l'étale sur des centaines ou milliers de ports : H grimpe fortement. Dans les deux cas, la répartition s'écarte de la valeur habituelle H_0 : c'est l'écart, dans un sens ou dans l'autre, qui donne l'alerte.

Corrigé de l'activité 3 — Inégalités de niveau de vie

1. Quintiles : $3{,}2 + 5{,}1 = 8{,}3$; $6{,}3 + 7{,}3 = 13{,}6$; $8{,}2 + 9{,}2 = 17{,}4$; $10{,}3 + 11{,}7 = 22{,}0$; $13{,}9 + 24{,}9 = 38{,}8$. Divisés par 100,1 :

$$p \approx (0{,}0829\ ;\ 0{,}1359\ ;\ 0{,}1738\ ;\ 0{,}2198\ ;\ 0{,}3876)$$

2. En additionnant les termes $-p_i \log_2 p_i$:

$$H \approx 0{,}298 + 0{,}391 + 0{,}439 + 0{,}480 + 0{,}530 \approx 2{,}138 \text{ bits}$$

3. $\log_2 5 = \frac{\ln 5}{\ln 2} \approx 2{,}322$, donc $I = 2{,}322 - 2{,}138 \approx 0{,}18$ bit (soit $I \times \ln 2 \approx 0{,}127$ pour l'indice de Theil des quintiles).

4. Si un seul groupe détenait tout : $H = 0$ et $I = \log_2 5 \approx 2{,}32$ bits (inégalité maximale). Si le revenu était parfaitement égal : $H = \log_2 5$ et $I = 0$. Plus I est grand, plus la répartition est inégale.

5. Avec les 10 déciles : $H \approx 3{,}111$ bits, $\log_2 10 \approx 3{,}322$ et $I \approx 0{,}21$ bit. L'écart est plus grand car regrouper les déciles en quintiles **masque** des inégalités à l'intérieur de chaque groupe (en particulier, le dernier décile détient à lui seul 24,9 %).

Corrigé de l'activité 4 — ADN

1. $\log_2 4 = 2$ bits, quand les 4 bases sont équiprobables ($g = \frac{1}{2}$).

2. $p_A = p_T = \frac{1-0{,}41}{2} = 0{,}295$ et $p_C = p_G = \frac{0{,}41}{2} = 0{,}205$. Alors :

$$H = -2(0{,}295\log_2 0{,}295 + 0{,}205\log_2 0{,}205) \approx 2(0{,}5195 + 0{,}4687) \approx 1{,}98 \text{ bit}$$

3. Comme $\log_2 \frac{x}{2} = \log_2 x - 1$:

$$H = -\left[(1-g)\log_2 \frac{1-g}{2} + g\log_2 \frac{g}{2}\right]$$

$$H = -[(1-g)\log_2(1-g) + g\log_2 g] + [(1-g) + g] = h(g) + 1$$

On reconnaît $H = h(g) + 1$.

4. $g = 0{,}20$: $H = 1 + h(0{,}2) \approx 1{,}72$ bit ; $g = 0{,}72$: $H \approx 1{,}86$ bit. Classement : Plasmodium (1,72) < Streptomyces (1,86) < humain (1,98) < 2. Plus le génome est déséquilibré (g loin de 0,5), plus l'entropie baisse.

5. $3{,}1 \times 10^9 \times 2 = 6{,}2 \times 10^9$ bits $= 7{,}75 \times 10^8$ octets, soit 775 Mo. À 1,98 bit par base : environ 766 Mo, soit une économie d'environ 9 Mo seulement (1 %) : l'ADN humain est presque incompressible base par base.

6. Dérivée :

$$h'(g) = \frac{-\ln g + \ln(1-g)}{\ln 2} = \log_2 \frac{1-g}{g}$$

Elle est positive pour $g < \frac{1}{2}$, nulle en $\frac{1}{2}$, négative ensuite. h est maximale en $g = \frac{1}{2}$ avec $h = 1$, donc H est maximale (2 bits) pour un génome à 50 % de GC.

Corrigé de l'activité 5 — Aléa : la faille Debian/OpenSSL

1. $32\,768 = 2^{15}$ valeurs équiprobables : $H = \log_2 2^{15} = 15$ bits.

2. Perte : $128 - 15 = 113$ bits. Fraction de l'espace réellement utilisée :

$$\frac{2^{15}}{2^{128}} = 2^{-113} \approx 9{,}6 \times 10^{-35}$$

3. Clés faibles : 32 768 s ≈ 9,1 heures. Clés d'un bon générateur : $2^{128} \approx 3{,}4 \times 10^{38}$ s, soit :

$$\frac{3{,}4 \times 10^{38}}{3{,}16 \times 10^7} \approx 1{,}1 \times 10^{31} \text{ ans}$$

Cela représente environ 8×10^{20} fois l'âge de l'Univers.

4. $\frac{32\,768}{10^6} \approx 0{,}03$ seconde : un attaquant énumère toutes les clés faibles instantanément.

5. La sécurité dépend de l'**entropie réelle** de la source d'aléa, pas de la longueur de la clé : une clé de 128 bits tirée parmi 32 768 valeurs n'offre que 15 bits de sécurité. C'est pourquoi les mots de passe choisis par les humains ou les clés mal générées sont si faibles.

Corrigé de l'activité 6 — Imagerie

1. $\frac{4+4\times2+4\times1}{16} = \frac{16}{16} = 1$. Image nette : $p = 1$ sur un pixel, $H = 1 \times \log_2 1 = 0$.

2. Poids : un pixel à $\frac{4}{16}$, quatre à $\frac{2}{16}$, quatre à $\frac{1}{16}$:

$$H_3 = \frac{4}{16}\log_2 4 + 4 \times \frac{2}{16}\log_2 8 + 4 \times \frac{1}{16}\log_2 16 = \frac{1}{2} + \frac{3}{2} + 1 = 3 \text{ bits}$$

3. $\log_2 9 \approx 3{,}17$. $H_3 = 3$ est inférieure car les poids ne sont pas égaux : le pixel central garde plus d'intensité que les coins.

4. On développe :

$$H(1/16, 4/16, 6/16, 4/16, 1/16) = 2 \times 1/16 \times 4 + 2 \times 1/4 \times 2 + 3/8 \log_2 8/3$$

$$\approx 0{,}5 + 1 + 0{,}531 = 2{,}031$$

Donc $H_5 \approx 4{,}06$ bits (maximum possible : $\log_2 25 \approx 4{,}64$).

5. $0 < 3 < 4{,}06$: plus l'image est floue, plus l'intensité est étalée et plus l'entropie est grande. Pour retrouver l'image la plus nette, l'algorithme cherche donc le réglage (ou la correction) qui **minimise** l'entropie.

Corrigé de l'activité 7 — Jeux

1. n questions distinguent au plus 2^n objets. Pour un million :

$$2^n \geq 10^6 \Leftrightarrow n \geq \frac{\ln 10^6}{\ln 2} \approx 19{,}93$$

Donc $n = 20$ questions ($2^{20} = 1\,048\,576$).

2. $2^6 = 64 < 100 \leq 128 = 2^7$: 7 questions. Stratégie : à chaque fois, couper l'ensemble en deux moitiés égales (dichotomie).

3. $h(1/2) = 1$ bit, alors que :

$$h(1/100) = 1/100 \log_2 100 + 99/100 \log_2 100/99 \approx 0{,}08 \text{ bit}$$

La question qui coupe en deux est 12 fois plus informative.

4. Information nécessaire :

$$\log_2 2\,315 = \frac{\ln 2315}{\ln 2} \approx 11{,}18 \text{ bits}$$

5. $3^5 = 243$ motifs, donc au plus $\log_2 243 \approx 7{,}92$ bits par essai. Comme $\frac{11{,}18}{7{,}92} \approx 1{,}41 > 1$, il faut au moins **2 essais** (et en pratique davantage, car les motifs ne sont jamais équiprobables).

Corrigé de l'activité 8 — Langues

1. $\log_2 26 \approx 4{,}70$ bits ; $\log_2 27 \approx 4{,}75$ bits.

2. Tableur : $H_0 \approx 4{,}01$ bits. Écart relatif : $1 - \frac{4{,}01}{4{,}70} \approx 15\ \%$: l'inégalité des fréquences (E à 17,8 %, W à 0,04 %) réduit déjà l'entropie.

3. $R = 1 - \frac{1{,}3}{4{,}75} \approx 0{,}73$ et $R = 1 - \frac{0{,}6}{4{,}75} \approx 0{,}87$: l'anglais est redondant à 73 % au moins, jusqu'à 87 %.

4. $\frac{12\,148 \times 8}{35\,149} \approx 2{,}77$ bits par caractère : bien en dessous de 8 (ASCII) et de l'entropie « lettre par lettre » du texte (4,57), mais encore au-dessus de l'estimation de Shannon (0,6 à 1,3) : un compresseur n'exploite qu'une partie de la structure de la langue.

5. Une langue utilise une toute petite partie des suites de lettres possibles : le contexte rend certaines lettres presque certaines (très faible quantité d'information), donc on peut reconstituer une lettre manquante ou repérer une faute.

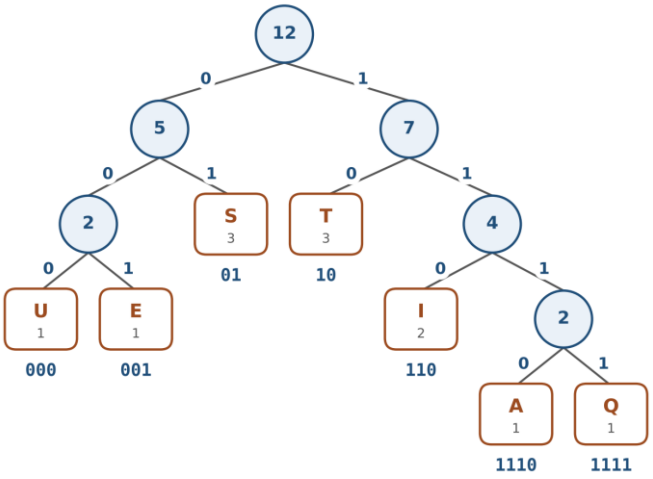
Corrigé de l'activité 9 — Compression : STATISTIQUES

1. On calcule :

$$H = 2 \times \frac{3}{12}\log_2 4 + \frac{2}{12}\log_2 6 + 4 \times \frac{1}{12}\log_2 12 = 1 + 0{,}431 + 1{,}195 \approx 2{,}63 \text{ bits}$$

2. On fusionne à chaque étape les deux plus petits poids :

Étape	Poids présents à la fin de l'étape	Fusion effectuée
0	S 3 ; T 3 ; I 2 ; A 1 ; Q 1 ; U 1 ; E 1	—
1	S 3 ; T 3 ; I 2 ; AQ 2 ; U 1 ; E 1	A + Q → 2
2	S 3 ; T 3 ; I 2 ; AQ 2 ; UE 2	U + E → 2
3	S 3 ; T 3 ; UE 2 ; IAQ 4	I + AQ → 4
4	T 3 ; IAQ 4 ; UES 5	UE + S → 5
5	UES 5 ; TIAQ 7	T + IAQ → 7
6	racine 12	UES + TIAQ → 12



3. S = 01, T = 10, I = 110, U = 000, E = 001, A = 1110, Q = 1111. Aucun code n'est le début d'un autre (toutes les lettres sont des feuilles).

4. Longueur moyenne :

$$L = \frac{3 \times 2 + 3 \times 2 + 2 \times 3 + 1 \times 3 + 1 \times 3 + 1 \times 4 + 1 \times 4}{12} = \frac{32}{12} \approx 2{,}67 \text{ bits}$$

On a bien $2{,}63 \leq 2{,}67 < 3{,}63$. Contrôle : $2 + 2 + 4 + 5 + 7 + 12 = 32$.

5. Huffman : 32 bits. Longueur fixe : $12 \times 3 = 36$ bits (gain $\frac{36-32}{36} \approx 11\ \%$). ASCII : $12 \times 8 = 96$ bits (gain $\frac{96-32}{96} \approx 67\ \%$).

Corrigé de l'activité 10 — Télécoms

1. $10^{35/10} = 10^{3,5} \approx 3\,162$; $10^{30/10} = 10^3 = 1\,000$.

2. Modem : $3\,100 \times \log_2(3\,163) \approx 36\,000$ bit/s. ADSL2+ : $2{,}2 \times \log_2(1\,001) \approx 21{,}9$ Mbit/s. Wi-Fi : $20 \times \log_2(1\,001) \approx 199$ Mbit/s.

3. 33,6 kbit/s pour environ 36 kbit/s possibles : le modem V.34 atteint 93 % de la limite de Shannon.

4. On résout $2{,}2\log_2\left(1+\frac{S}{N}\right) = 24$:

$$\log_2\left(1+\frac{S}{N}\right) = \frac{24}{2{,}2} \approx 10{,}91$$

$$1+\frac{S}{N} = 2^{10{,}91} \approx 1\,922 \quad \text{donc} \quad \frac{S}{N} \approx 1\,921 \quad \text{soit} \quad 10\log_{10}(1\,921) \approx 32{,}8 \text{ dB}$$

Le débit maximal n'est donc possible que si le signal est assez fort : sur une **ligne courte**, proche du central ; sur une ligne longue, le signal est atténué et le débit s'effondre.

5. Doubler B : $C' = 2B\log_2\left(1+\frac{S}{N}\right) = 2C$. Doubler $\frac{S}{N}$: $B\log_2(1+2000) \approx B \times 10{,}97$ contre $B \times 9{,}97$, soit seulement $+B$ bit/s, car $\log_2$ croît lentement. D'où l'intérêt d'élargir les canaux (20, 40, 80, 160 MHz). Le débit annoncé du Wi-Fi (72,2 Mbit/s) reste très inférieur à 199 Mbit/s : les modulations réelles ont un nombre fini de niveaux et gardent des marges contre le bruit.

Corrigé de l'activité 11 — Mots de passe

1. $H_1 = 8\log_2 26 \approx 37{,}6$ bits, soit $26^8 \approx 2{,}1 \times 10^{11}$ combinaisons. $H_2 = 16\log_2 26 \approx 75{,}2$ bits, $26^{16} \approx 4{,}4 \times 10^{22}$. $H_3 = 8\log_2 94 \approx 52{,}4$ bits, $94^8 \approx 6{,}1 \times 10^{15}$.

2. NTLM (3×10^{11} essais par seconde) :

$$\text{①}\ \frac{2{,}1 \times 10^{11}}{3 \times 10^{11}} \approx 0{,}7 \text{ s}$$

$$\text{②}\ \frac{4{,}4 \times 10^{22}}{3 \times 10^{11}} \approx 1{,}5 \times 10^{11} \text{ s} \approx 4\,600 \text{ ans}$$

$$\text{③}\ \frac{6{,}1 \times 10^{15}}{3 \times 10^{11}} \approx 2{,}0 \times 10^{4} \text{ s} \approx 5{,}6 \text{ heures}$$

3. bcrypt (2×10^5 essais par seconde) :

$$\text{①}\ \frac{2{,}1 \times 10^{11}}{2 \times 10^{5}} \approx 1{,}0 \times 10^{6} \text{ s} \approx 12 \text{ jours}$$

$$\text{③}\ \frac{6{,}1 \times 10^{15}}{2 \times 10^{5}} \approx 3{,}0 \times 10^{10} \text{ s} \approx 970 \text{ ans}$$

L'algorithme d'empreinte, 1,5 million de fois plus lent, change tout : un mot de passe de 8 caractères complexes passe de quelques heures à des siècles.

4. Pour NTLM, ② résiste 4 600 ans contre 5,6 heures pour ③, pourtant « plus compliqué » en apparence. Dans $H = L\log_2 N$, la longueur L multiplie l'entropie alors que N n'intervient que par son logarithme : ajouter des caractères est bien plus efficace que diversifier l'alphabet, comme le recommande le NIST.

5. $p = \frac{290\,731}{32\,600\,000} \approx 0{,}0089$ (0,9 %), donc quantité d'information $\log_2 \frac{1}{p} \approx 6{,}8$ bits, contre 37,6 bits pour un mot de passe ① tiré au hasard. Les mots de passe humains sont très prévisibles : l'attaquant essaie d'abord les plus courants.

Corrigé de l'activité 12 — Intelligence artificielle : le Titanic

1. $p = \frac{711}{2201} \approx 0{,}323$:

$$H_{\text{total}} = -(0{,}323\log_2 0{,}323 + 0{,}677\log_2 0{,}677) \approx 0{,}526 + 0{,}381 \approx 0{,}908 \text{ bit}$$

2. Femmes ($p = 0{,}732$) : $H_F \approx 0{,}839$ bit. Hommes ($p = 0{,}212$) : $H_H \approx 0{,}745$ bit.

3. Moyenne pondérée par les effectifs :

$$H_{\text{sexe}} = \frac{470}{2201} \times 0{,}839 + \frac{1731}{2201} \times 0{,}745 \approx 0{,}179 + 0{,}586 \approx 0{,}765 \text{ bit}$$

$$G_{\text{sexe}} \approx 0{,}908 - 0{,}765 \approx 0{,}142 \text{ bit}$$

4. 1re classe ($p = 0{,}625$) : $H \approx 0{,}955$; autres ($p = 0{,}271$) : $H \approx 0{,}843$. Donc :

$$H_{\text{classe}} = \frac{325}{2201} \times 0{,}955 + \frac{1876}{2201} \times 0{,}843 \approx 0{,}859 \text{ bit}$$

$$G_{\text{classe}} \approx 0{,}908 - 0{,}859 \approx 0{,}049 \text{ bit}$$

5. On compare les gains :

$$G_{\text{sexe}} \approx 0{,}142 > G_{\text{classe}} \approx 0{,}049$$

L'arbre interroge d'abord le **sexe**. C'est cohérent avec la consigne « les femmes et les enfants d'abord » : elle a pesé plus lourd sur la survie que la classe du billet, même si la classe a joué (accès aux canots).

Corrigé de l'activité 13 — Corriger les erreurs

1. Le bit décodé est faux si au moins 2 copies sur 3 sont erronées :

$$P(X \geq 2) = \binom{3}{2}(0{,}1)^2(0{,}9) + (0{,}1)^3 = 0{,}027 + 0{,}001 = 0{,}028$$

2. Au moins 3 copies erronées sur 5 :

$$P(X \geq 3) = \binom{5}{3}(0{,}1)^3(0{,}9)^2 + \binom{5}{4}(0{,}1)^4(0{,}9) + (0{,}1)^5 = 0{,}0081 + 0{,}00045 + 0{,}00001 \approx 0{,}0086$$

3. Rendements : $\frac{1}{3} \approx 0{,}33$ et $\frac{1}{5} = 0{,}2$ bit utile par bit envoyé : on gagne en fiabilité mais on perd en débit.

4. Sans protection : $E = 1\,000 \times 0{,}1 = 100$ bits faux. Avec le code par 3 : $E = 1\,000 \times 0{,}028 = 28$ bits faux (mais 3 000 bits envoyés).

5. $C = 1 - h(0{,}1) \approx 1 - 0{,}469 \approx 0{,}53$. Il existe donc, d'après Shannon, des codes de rendement proche de 0,53 avec une erreur arbitrairement faible : le code à répétition par 3 (rendement 0,33) est loin d'être optimal. Les codes réels (QR codes, CD, sondes spatiales) sont bien plus efficaces.

6. $C = 0 \Leftrightarrow h(p) = 1 \Leftrightarrow p = \frac{1}{2}$: si un bit sur deux est inversé, la sortie est indépendante de l'entrée et le canal ne transmet **aucune** information.