Chiffre de Vigenère

Ce rapport résume l'ensemble du travail réalisé pour casser un chiffre de Vigenère encodant les 26 lettres plus l'espace (alphabet = A..Z + , modulo 27).

Logique et méthode :

Nous avons commencé par estimer la longueur de la clé avec Kasiski et l'indice de coïncidence : les résultats pointaient vers une petite période (facteurs 2 et 4). Ensuite nous avons testé deux méthodes pour retrouver la clé : une méthode rapide qui maximise le nombre d'espaces reconstitués, et une méthode statistique (χ^2) qui compare la distribution des lettres au français attendu.

1- Nettoyage — on lit vigenere.txt, on met en majuscules et on conserve seulement A–Z et l'espace (travail en modulo 27).

2- Estimation de la longueur de clé — on applique :

   a- **Kasiski** (recherche de répétitions et factorisation des distances) → candidates pour k,

   b- **Friedman** (indice de coïncidence) → confirmation statistique.

3- Recherche de la clé — deux approches parallèles :

   a- **Méthode "spaces"** : pour chaque colonne modulo k, on teste chaque shift et on choisit celui qui maximise le nombre d'espaces (simple, rapide).

   b- **Méthode χ^2 (retenue) :** pour chaque colonne on prend le shift minimisant le χ^2 entre la distribution observée (après shift) et la

distribution attendue du français (A..Z + espace). C'est la méthode statistique robuste.

4- Validation croisée — on compare résultats spaces / χ² et on choisit la solution la plus cohérente (ici χ² → TUPQ).

5- Déchiffrement — appliquer la clé retrouvée pour produire le texte clair.

Résultat principal :

clé = TUPQ (k = 4) .

fichier déchiffré : vigenere_dechiffre_by_chi2.txt.

```
Summary (diagnostics):

Len cipher (cleaned) = 4878
IC (friedman) = 0.04778
Kasiski factors (top 12): [(2, 4930), (4, 4871), (8, 2296), (3, 1549), (6, 1509), (12, 1495), (16, 1069), (5, 874), (10, 853), (20, 841), (24, 661), (7, 619)]

Testing k=1..20 (spaces method & chi2 method)
k= 1  spaces_score=  475 key_spaces=U   chi2_score=12820.7 key_chi2=D
k= 2  spaces_score=  528 key_spaces=YZ   chi2_score=12579.9 key_chi2=DE
k= 3  spaces_score=  475 key_spaces=UUU   chi2_score=12935.8 key_chi2=DDD
k= 4  spaces_score=  775 key_spaces=TUPQ   chi2_score=230.3 key_chi2=TUPQ
k= 5  spaces_score=  475 key_spaces=UUUUU   chi2_score=13325.7 key_chi2=DDDDD
k= 6  spaces_score=  542 key_spaces=YZUVYZ   chi2_score=12600.5 key_chi2=DHDEGE
k= 7  spaces_score=  475 key_spaces=UUUUUUU   chi2_score=13003.6 key_chi2=DHDDDDD
k= 8  spaces_score=  783 key_spaces=TUPQYUPV   chi2_score=334.3 key_chi2=TUPQTUPQ
k= 9  spaces_score=  475 key_spaces=UUUUUUUUU   chi2_score=13086.7 key_chi2=DHDDDDHHH
k=10  spaces_score=  561 key_spaces=YZUVYVYZUV   chi2_score=12301.3 key_chi2=PEDEDEGHDH
k=11  spaces_score=  482 key_spaces=HUUUUUUUUUU   chi2_score=13484.2 key_chi2=DPDHDDDDGDD
k=12  spaces_score=  803 key_spaces=TUPQTZPQYUPV   chi2_score=409.0 key_chi2=TUPQTUPQTUPQ
k=13  spaces_score=  481 key_spaces=UYQUUUUUUUUUU   chi2_score=13460.3 key_chi2=PDDDHDHDHDHDD
k=14  spaces_score=  561 key_spaces=YVPZPUYVYZUZUZ   chi2_score=11927.5 key_chi2=DHPEPEDEGHGEDE
k=15  spaces_score=  477 key_spaces=TUUUUUUUUUUUUUU   chi2_score=12819.9 key_chi2=PQDDDHDDDDPDHHD
k=16  spaces_score=  793 key_spaces=TUPQTUPVYUPQYZPV   chi2_score=548.1 key_chi2=TUPQTUPQTUPQTUPQ

Best by spaces: k=20 key=TZPVTUPQYUPQYUUQYZPQ
Best by chi2  : k=4 key=TUPQ
```

La sortie (Best by spaces: k=20 ...) signifie que la méthode des espaces a obtenu son score maximal pour k=20.
Ceci s'explique parce que k=20 est un multiple de la vraie période (4) : les colonnes pour k=20 agrègent plusieurs phases de la clé et peuvent, globalement, produire plus d'espaces.
Cette agrégation crée une illusion statistique — beaucoup d'espaces mais des lettres qui ne correspondent plus à la langue.
La méthode χ², en revanche, compare la distribution complète des caractères au profil attendu du français et n'est pas sensible à cette agrégation.
Dans notre diagnostic χ² atteint un minimum clair pour k=4 avec la clé **TUPQ** (χ² ≈ 230 contre χ² ≈ 667 pour k=20), ce qui indique une meilleure qualité linguistique.

Le déchiffrement avec **TUPQ** produit un texte lisible ; c'est la preuve empirique qui confirme la supériorité de la solution χ^2 ici.
En pratique, on privilégie χ^2 lorsque *spaces* pointe un multiple de période : vérifier la valeur χ^2 et relire un extrait du texte déchiffré permet de trancher..

Code Source :

Le dépôt Git associé est disponible à : https://github.com/ThemlaouiHou/travail-pratique-1-/tree/main

Le README de l'exercice contient les instructions pour lancer le script et visualiser les résultats.

LLM (usage réel) :

L'IA (ChatGPT) a servi uniquement à proposer un squelette de pipeline et des idées de mise en œuvre — toute la validation, l'implémentation, l'exécution et l'interprétation ont été réalisées par l'équipe.

- Prompt principal utilisé (extrait exact)

" - Propose un pipeline Python pour casser un chiffre de Vigenère (alphabet A–Z + espace).

- Inclue Kasiski, Friedman, méthode χ^2 et une méthode par maximisation des espaces.
- Fournis scripts prêts à exécuter. "