Devoir : Statistiques, séries et dénombrement

Le sujet se compose d'un exercice et de deux problèmes. Dans ces derniers, les parties ne sont pas de difficultés croissantes et on prendra soin de lire l'ensemble du sujet avant de commencer à composer. (A titre indicatif, on pourra consacrer une demi-heure à l'exercice, une heure au Problème 1 et une heure et demi au Problème 2).

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il convient de le signaler sur la copie et de poursuivre la composition en expliquant les raisons des initiatives qui ont été prises.
Il sera tenu compte de la présentation et en particulier de $\boxed{\textit{l'encadrement}}$ *des résultats.*
L'usage de la calculatrice ***est*** *autorisé au cours de l'épreuve. Les candidats pourront admettre le résultat d'une question pour répondre à une question postérieure à condition de le mentionner explicitement.*

Exercice : une formule combinatoire

On souhaite montrer l'égalité : $\displaystyle\sum_{k=a}^{n} \binom{k}{a} = \binom{n+1}{a+1}$.

① *Démonstration combinatoire.* On considère une urne U contenant $n+1$ boules dont b sont noires et $a+1$ sont blanches. On extrait les boules une à une jusqu'à vider l'urne.

a) Combien de tirages différents sont possibles ?

b) Soit M_k l'événement : « La dernière boule blanche occupe la $(k+1)$-ième place ». Déterminer $\mathrm{Card}(M_k)$ pour des valeurs de k qu'on précisera.

c) En déduire la formule annoncée.

② Fournir la démonstration algébrique de cette égalité par télescopages.

③ *Modélisation*

a) Écrire une fonction Python `simulTirage(N,A)` de paramètres d'entrée le nombres `N` ($N = n+1$) de boules dans l'urne et le nombre `A` de blanches ($A = a+1$), et qui modélise le résultat d'un tel tirage en retournant une liste formée de 0 et de 1, selon que la boule est noire ou blanche.

b) Écrire une fonction `rangDerniereBlanche(L)` qui retourne k si la dernière boule blanche occupe la $(k+1)$-ième place (les places étant numérotées de 1 à $n+1$ conformément à la question 1.).

c) Soit A_k l'événement : « la dernière boule blanche occupe la $(k+1)$-ième place ». Que vaut $\mathbb{P}(A_k)$?
Écrire une fonction `estimProba(N,A,k,m)` d'arguments le nombre $N = n+1$ de boules dans l'urne, le nombre $A = a+1$ de blanches, l'indice `k` de l'événement et un entier `m` supposé grand (de l'ordre du millier) qui retourne une estimation de $\mathbb{P}(A_k)$, en faisant appel un grand nombre de fois à la fonction `simulTirage()`.

Problème 1 : Séries numériques

1. Étude de la somme d'une série

Pour tout entier naturel $n \geq 1$, on pose :

$$S_n = \sum_{k=1}^{n} \frac{1}{k^2}$$

① Quel est le résultat de cours qui permet d'affirmer que la suite (S_n) converge ? Le montrer.

Dans la suite du problème, on notera $l = \lim\limits_{n \to \infty} S_n$.

② Écrire une fonction Python `sommeP(n)` qui prend en argument n et qui renvoie la valeur de S_n.

③ a) Soit $f : t \longmapsto \dfrac{1}{t^2}$.
Pour tout entier $k \geq 2$, montrer que $\forall x \in [k, k+1]$, $f(t) \leq f(k)$ et en déduire que $\displaystyle\int_k^{k+1} \frac{1}{t^2} dt \leq \frac{1}{k^2}$.
Montrer plus généralement que pour tout entier $k \geq 2$, on a :

$$\int_k^{k+1} \frac{1}{t^2} dt \leq \frac{1}{k^2} \leq \int_{k-1}^{k} \frac{1}{t^2} dt$$

b) Soit p et n deux entiers naturels non nuls.
Démontrer que :

$$\int_{n+1}^{n+p+1} \frac{1}{t^2} dt \leq S_{n+p} - S_n \leq \int_n^{n+p} \frac{1}{t^2} dt$$

c) Démontrer alors que :

$$\frac{1}{n+1} \leq l - S_n \leq \frac{1}{n}$$

puis l'équivalent :

$$l - S_n \underset{n \to \infty}{\sim} \frac{1}{n}$$

Que peut-on dire de $S_n + \dfrac{1}{n}$ pour n suffisamment grand ?

④ On considère la fonction suivante :

```
def estimation(n):
   s = sommeP(n)
   s = sqrt(6*(s+1/n))
   return (s)
```

L'instruction `estimation(1000)` renvoie la valeur : 3.141593130895433.
Que peut-on conjecturer quant à la probable valeur de l ?

2. Étude de la nature d'une série

Pour tout entier naturel $n \geq 1$, on pose :

$$u_n = \prod_{k=1}^{n} \left(1 + \frac{1}{k^2}\right) = (1+1)\left(1 + \frac{1}{4}\right) \cdots \left(1 + \frac{1}{n^2}\right)$$

① Donner, sous forme d'entiers ou de fractions simplifiées, les valeurs de u_1, u_2 et u_3.

② a) Montrer que, pour tout entier naturel $n \geq 1$, on a : $u_n \geq 2$.
b) Étudier la monotonie de la suite (u_n).

③ a) Montrer que, pour tout réel x strictement supérieur à -1, on a : $\ln(1+x) \leq x$.
b) Démontrer alors que, pour tout $n \in \mathbb{N}^*$, on a :

$$\ln(u_n) \leq l \qquad \text{(où } l = \sum_{k=1}^{\infty} \frac{1}{k^2} \text{ a été estimée dans la partie 1.)}$$

④ En utilisant les questions précédentes, montrer que la suite (u_n) converge vers un réel l', élément de $[2, e^l]$.

⑤ On se propose dans cette question de déterminer la nature de la série de terme général $l' - u_n$.

a) Justifier que la suite $(\ln(u_n))_{n \in \mathbb{N}^*}$ converge et que l'on a :

$$\ln(l') = \sum_{k=1}^{+\infty} \ln\left(1 + \frac{1}{k^2}\right)$$

b) Montrer que, pour tout $n \in \mathbb{N}^*$, on a :

$$\ln\left(\frac{l'}{u_n}\right) = \sum_{k=n+1}^{+\infty} \ln\left(1 + \frac{1}{k^2}\right)$$

c) Écrire en Python une fonction `reste` qui prend en argument n et qui renvoie une bonne valeur approchée de :

$$\sum_{k=n+1}^{\infty} \ln\left(1 + \frac{1}{k^2}\right)$$

d) Voici les résultats de quelques appels de cette fonction (supposée correctement écrite) :

```
> > > reste(9)
0.10497285546814518

> > > reste(49)
0.020198959583845037

> > > reste(99)
0.010048997486379833

> > > reste(199)
0.0050114998428795786

> > > reste(499)
0.0020009999955394315

> > > reste(999)
0.00099949999929165772
```

Que peut-on alors conjecturer comme équivalent en l'infini très simple de $\displaystyle\sum_{k=n+1}^{\infty} \ln\left(1 + \frac{1}{k^2}\right)$?

e) En admettant le résultat conjecturé dans la question précédente, justifier qu'il existe un entier naturel n_0 (qu'on ne cherchera pas à calculer) tel que, pour tout $n \geq n_0$, on a :

$$\sum_{k=n+1}^{\infty} \ln\left(1 + \frac{1}{k^2}\right) \geq \frac{0,9}{n}$$

f) En déduire que pour tout $n \geq n_0$, on a :

$$l' - u_n \geq u_n(e^{0,9/n} - 1)$$

g) On admet que pour tout réel x, on a : $e^x - 1 \geq x$.
Démontrer alors que :

$$\forall n \geq n_0,\ l' - u_n \geq \frac{1,8}{n}$$

Conclure quant à la nature de la série de terme général $l' - u_n$.

Problème 2 : Agro-Véto B 2018

L'équation de Michaelis-Menten permet de décrire la cinétique d'une réaction catalysée par une enzyme agissant sur un substrat unique pour donner un produit. Elle relie la vitesse de la réaction à la concentration de substrat et à des paramètres constants, caractéristiques de l'enzyme.

D'après les travaux de Michaelis et Menten, lors d'une réaction chimique, un substrat, noté S, est catalysé par une enzyme, notée E, pour obtenir un produit, noté P. Plus précisément, le substrat se fixe sur l'enzyme pour former un complexe transitoire appelé « enzyme-substrat », noté C, qui se décompose ensuite pour donner le produit et l'enzyme, selon la réaction suivante :

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} C \xrightarrow{k_2} P + E$$

On considère que la vitesse de chacune des réactions est proportionnelle au produit des concentrations des réactifs. Les constantes k_1, k_{-1} et k_2 désignent les constantes de vitesse associées aux différentes réactions et sont supposées strictement positives. Dans la suite, et pour simplifier les écritures, on notera les différentes concentration (exprimées en $mol.L^{-1}$) par des lettres minuscules :

$$s = [S], \ e = [E] \text{ et } p = [P],$$

en remarquant que les concentrations sont des fonctions dépendant du temps.
On complète cette introduction en précisant les conditions initiales, à savoir :

$$s(0) = s_0, \ e(0) = e_0, \ c(0) = 0 \text{ et } p(0) = 0 \text{ avec } s_0 > 0 \text{ et } e_0 > 0.$$

L'objectif de cette partie est d'étudier la vitesse $\dfrac{dp}{dt}$ à laquelle le produit se forme en admettant que :

$$\frac{dp}{dt}(t) = \frac{v_{max}s(t)}{K_M + s(t)} \text{ où } K_M = \frac{k_2 + k_{-1}}{k_1} > 0 \text{ et } v_{max} = k_2 e_0 > 0.$$

1.1. Étude du modèle

On considère l'équation suivante, appelée équation de Michaelis-Menten :

$$\frac{dp}{dt} = \frac{v_{max}s}{K_M + s}$$

Pour étudier cette équation, on définit sur $\mathbb{R}_+$ la fonction de Michaelis-Menten, notée f, par :

$$f(s) = \frac{v_{max}s}{K_M + s}$$

① Montrer que la fonction f est croissante et déterminer ses limites aux bords du domaine de définition.

② Pour quelle valeur de s a-t-on $f(s) = \dfrac{v_{max}}{2}$?

③ Tracer la courbe représentative de f et y faire figurer des informations pertinentes (comme par exemple la tangente en 0).

1.2. Identification expérimentale des paramètres

Dans la suite, la quantité $\dfrac{dp}{dt}(t)$ est notée $v(t)$. On note par ailleurs v_i la vitesse initiale et on obtient :

$$v_i = \frac{v_{max}s_0}{K_M + s_0}$$

On utilise dorénavant une approche, mise au point par Lineweaver et Burk, afin de déterminer expérimentalement K_M et v_{max}.

① Établir une relation de la forme $v_i^{-1} = \alpha s_0^{-1} + \beta$ où les constantes α et β sont à déterminer.

② Expliquer comment on peut déterminer graphiquement les paramètres K_M et v_{max} à partir des données expérimentales.

On se propose d'appliquer l'approche précédente sur des résultats expérimentaux de Michaelis et Menten concernant l'hydrolyse de saccharose sous l'action d'une enzyme, l'invertase. Le tableau suivant donne les vitesses initiales en fonction des concentrations initiales en saccharose pour 7 expérimentations, ainsi que leurs inverses arrondis à l'unité.

N° d'expérience	s_0 (en $mol.L^{-1}$)	v_i (en $mol.L^{-1}.min^{-1}$)	s_0^{-1}	v_i^{-1}
1	0.330	$3.636.10^{-3}$	3	275
2	0.1670	$3.636.10^{-3}$	6	275
3	0.0833	$3.236.10^{-3}$	12	309
4	0.0416	$2.666.10^{-3}$	24	375
5	0.0208	$2.114.10^{-3}$	48	473
6	0.0104	$1.466.10^{-3}$	96	682
7	0.0052	$0.866.10^{-3}$	192	1155

③ Reporter sur le graphique de l'annexe 1 les couples (s_0^{-1}, v_i^{-1}).

④ Proposer des valeurs approchées de K_M et v_{max} avec une précision en accord avec l'approche utilisée.

1.3. Étude informatique de données expérimentales

✐ *Remarque :* Les extraits de code matérialisés par des `---` correspondent à des portions à compléter.
Dans cette partie, on suppose que le fichier Python débute par l'importation du modyle `matplotlib.pyplot` et par la définition des deux listes `Ls` et `Lv`, de la façon suivante :

```
import matplotlib.pyplot as plt
Ls = [0.333,0.167,0.0833,0.0416,0.0208,0.0104,0.0052]
Lv = [3.636,3.636,3.236,2.666,2.114,1.466,0.866]
```

① a) Écrire une fonction `inv` qui prend en entrée une liste de nombres (supposés non nuls) `L` et qui renvoie la liste composée des inverses de ces nombres. *Exemple :* `inv([0.25,2,1,8])` renvoie `[4.0,0.5,1.0,0.125]`.
b) Écrire une version améliorée `inv_ex` de la fonction `inv` qui prend en entrée une liste de nombres `L`, puis : si un de ces nombres est nul, alors elle renvoie le booléen `False`, sinon elle renvoie la liste composée des inverses de ces nombres.
c) Quelle ligne de code écrire en Python afin qu'elle effectue le tracé demandé en question 3 de la partie 1.2 (les points seront représentés par des petits cercles et ne seront pas reliés entre eux).

② Écriture de fonctions préliminaires. *Pour cette question, on s'interdit d'utiliser les commandes préprogrammées de Python qui renvoient la somme, la moyenne ou la variance. Avant chaque fonction, on écrira brièvement le raisonnement suivi et la formule qu'elle est censée calculer.*

a) Écrire une fonction `moyenne` qui prend en entrée une liste `X` (non vide) de nombres réels et qui renvoie la moyenne des éléments de la liste.
b) Écrire une fonction `variance` qui prend en entrée une liste `X` (non vide) de nombres réels et qui renvoie la variance des éléments de la liste.

③ a) Compléter le programme suivant afin qu'il renvoie la valeur de la covariance de `X` et `Y` si elle existe et le booléen `False` sinon (*mêmes consignes qu'à la question précédente*).

```
def cov(X,Y):
   " " " Entree : X,Y (liste). " " "
   nx = len(X) ; ny = len(Y)
   if ----- or nx == 0:
      return(False)
   else:
      -------
      for k in range(------):
         S = ------
      y = 1/nx*S
      return(y)
```

b) Parmi les quatre valeurs suivantes, lesquelles ne peuvent pas être renvoyées par la fonction `cov` ? (*on justifiera les réponses*) :

i. `True`
ii. `[1,2]`
iii. `"False"`
iv. `-0.5`

c) On considère les fonctions `Coef` et `Trace` suivantes :

```
def Coef(X,Y):
   a = cov(X,Y)/variance(X)
   b = moyenne(Y)-cov(X,Y)/variance(X)*moyenne(X)
   return([a,b])

def Trace(X,Y):
   [a,b] = Coef(X,Y)
   xmin = min(X) ; xmax = max(X)
   plt.plot(X,Y,"*")
   plt.plot( ------- )
   plt.plot([moyenne(X)],[moyenne(Y)],"s")
   plt.grid()
   plt.show()
```

i. Compléter la fonction `Trace` afin de tracer le segment d'extrémités `(xmin,a*xmin+b)` et `(xmax,a*xmax+b)`.
ii. Donner l'équation de la droite qui passe par ces deux points.
iii. Quel nom porte cette droite.
iv. On exécute la fonction `Trace` pour des listes `X` et `Y` quelconques de taille 5. Pour chacun des trois tracés suivants, indiquer avec justification s'il peut être ou non le résultat de `Trace` ?

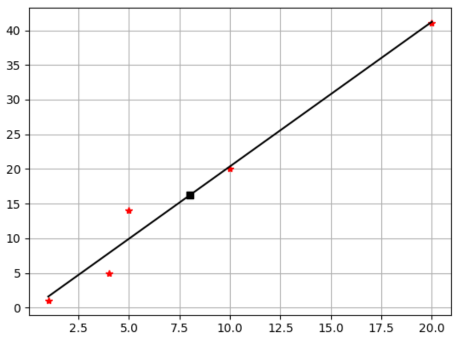

FIGURE 1 – (a) Tracé n°1

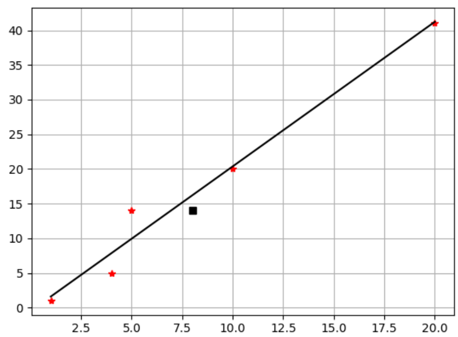

FIGURE 2 – (b) Tracé n°2

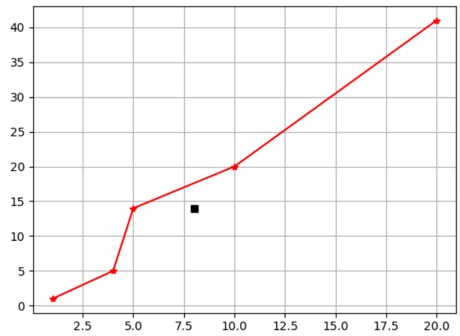

FIGURE 3 – (c) Tracé n°3

d) i. En utilisant les fonctions et variables qui précèdent, proposer un code qui calcule les valeurs de K_M et v_{max} en suivant la démarche de la partie 1.2

ii. Pour ces données, le coefficient de corrélation linéaire vaut 0.9995. Qu'en déduire ?

1.4. Analyse de l'équation de Michaelis-Menten par Schnelle et Mendoza

Les parties précédentes ont permis de déterminer expérimentalement les constantes K_M et v_{max}. Dans cette partie, on s'intéresse à la dépendance de s par rapport au temps, sous l'hypothèse de l'Approximation des Etats Quasi Stationnaires (AEQS) selon laquelle la variation de la concentration en complexe « enzyme-substrat » est nulle (car il est consommé par la réaction juste après sa création), à l'exception d'une très courte phase initiale de durée $\delta > 0$. On considère que la concentration en S ne varie pas au cours de cette phase initiale.

Sous cette hypothèse, pour tout $t \in [\delta, +\infty[$, on a $\dfrac{ds}{dt}(t) = -\dfrac{dp}{dt}(t) = -\dfrac{v_{max}s(t)}{K_M + s(t)}$.
On s'intéresse désormais à l'équation différentielle :

$$\frac{ds}{dt} = -\frac{v_{max}s}{K_M + s}$$

avec pour condition initiale $s(\delta) = s_0$.

① Soit g la fonction définie sur $\mathbb{R}_+$ par $g(x) = xe^x$.

a) Démontrer que g est strictement croissante.

b) En déduire que la fonction h, réciproque de g, est bien définie et donner son domaine de définition.

c) Tracer la courbe représentative de la fonction h en expliquant la méthode graphique utilisée.

② On définit $y(t) = g\left(\frac{s(t)}{K_M}\right)$. Écrire une équation différentielle du premier ordre satisfaite par la fonction y et donner la condition initiale correspondante.

③ Déduire des questions précédentes une expression de $y(t)$ puis de $s(t)$ en fonction de t, K_M, v_{max} et s_0 (et faisant appel à la fonction h).

④ Proposer une méthode numérique permettant d'approcher les valeurs de la fonction h.