RAPPORT D'ACTIVITÉ

Analyse de données – Parcours débutant

Par Anthéa BLIECK

Numéro étudiant : 21238624

Etudiante en première année du Master GAED-SCT

Faculté des Lettres, Sorbonne Université

Table des matières

Séance 1

I- Installation de Python

Après avoir installé Docker sur mon ordinateur personnel, et m'être connectée sur mon compte, j'ai téléchargé les fichiers nécessaires à l'installation de Python. Il s'est avéré que, sur mon ordinateur, cette démarche n'a pu fonctionner, même après demande d'aide à mon professeur. J'ai donc finalement installé Python en dur sur mon ordinateur. J'ai ainsi perdu plus de 2 cours à tenter de dépasser ces difficultés techniques.

Une fois Python et Notepad+ installés de manière fonctionnelle j'ai pu télécharger les fichiers suivants pour débuter la deuxième séance :

- requirements.txt
- main.py
- resultats-elections-presidentielles-2022-1er-tour.csv

Séance 2

I- Questions de cours

1. La géographie a longtemps eu une relation complexe avec la statistique. En tant que discipline des sciences humaines et sociales, elle a parfois fait passer la rigueur mathématique au second plan, alors même qu'elle produit ou utilise quantité de données dont une bonne compréhension, voire une décantation des idées reçues, ne saurait se passer. Il n'en demeure pas moins que la statistique est aujourd'hui incontournable en géographie, car elle permet non seulement de digérer, mais aussi de structurer et d'interpréter une partie de la masse des informations recueillies sur le terrain. En ce sens, on peut dire que la statistique ne remplace pas la réflexion géographique, mais qu'elle l'éclaire en proposant des outils de mesure, de synthèse ou de modélisation.

2. La géographie, en tant que science, s'efforce de limiter dans la mesure du possible la place du hasard dans les phénomènes qu'elle étudie. Les philosophes distinguent deux postures face au hasard. D'un côté les partisans du déterminisme, qui affirment que le hasard n'existe pas car, derrière chaque variation, il y a une cause qui la détermine. D'un autre côté, d'inspiration probabiliste, ceux qui estiment que le hasard existe mais qu'il résulte de causes encore inconnues. En géographie le hasard est souvent la marque de la contingence : on ne saurait déterminer avec exactitude et précision le comportement des individus ou des acteurs. Cependant, il est souvent possible d'en dégager des grandes

tendances. En cela, d'un point de vue géographique, le hasard n'empêche pas qu'on puisse avoir une connaissance scientifique du monde, il en trace seulement les limites.

3. L'information géographique s'articule autour de deux grandes catégories. La première regroupe des données attributaires, relatives à la géographie humaine (population, activités économiques, structure sociale) ou physique (température, reliefs…). La seconde catégorie correspond aux données géométriques, qui restituent la morphologie des territoires et des objets spatiaux. Quelle que soit leur nature, les données doivent être accompagnées de métadonnées garantissant leur fiabilité et leur traçabilité.

4. L'analyse des données en géographie fait appel à des compétences méthodologiques et à une véritable rigueur scientifique. Cette dernière repose sur la production ou la collecte d'informations, leur structuration puis leur traitement statistique. La géographie a donc recours aux statistiques pour explorer la structure interne des phénomènes spatiaux et confronter les résultats obtenus sur le terrain. Les statistiques utilisées par la géographie se subdivisent en deux grandes catégories : la statistique descriptive et la statistique explicative.

5. La statistique descriptive vise à présenter, résumer et ordonner les données observées. Elle repose sur le calcul de certains indicateurs (moyenne, médiane, écart-type) et sur la confection de représentations graphiques (histogrammes, diagrammes en secteurs, boîtes à moustaches). Elle permet de donner une vision simplifiée de la réalité et de préparer le terrain pour des analyses plus poussées.

La statistique explicative, elle, a pour objectif d'établir, d'étudier, de modéliser ou de quantifier des relations de cause à effet qui peuvent exister entre deux variables ou plus. Elle regroupe un ensemble de techniques permettant au chercheur d'établir un lien entre une variable dépendante et un ensemble de variables explicatives. Ce rapport peut être modélisé de manière plus ou moins complexe selon le nombre et la nature des variables, et selon le type de relation attendu. Par ce biais le statisticien ou le géographe peuvent ainsi chercher à comprendre, expliquer, prévoir ou encore modéliser les phénomènes spatiaux qu'il étudie.

6. Les visualisations utilisées en géographie varient selon la nature des données.
Pour les variables continues, on mobilise fréquemment des histogrammes, des polygones de fréquence, des courbes cumulatives décroissantes, des boîtes à moustaches ou encore des méthodes de lissage normal. Les variables discrètes sont souvent représentées par des diagrammes en bâtons, tandis que les variables qualitatives se prêtent plutôt aux diagrammes en secteurs (camemberts). D'autres formes de représentations existent également, comme les diagrammes à rectangles horizontaux, les polygones ou encore les visualisations de proximités issues d'analyses factorielles.

Le choix d'un type de visualisation dépend à la fois du type de variable (qualitative, quantitative, discrète, continue) et de l'objectif analytique : mettre en évidence une distribution, comparer des groupes, détecter des valeurs atypiques ou représenter des proximités multivariées. Ainsi, un histogramme convient particulièrement pour examiner la forme d'une distribution continue, un diagramme en secteurs pour montrer la composition relative d'une variable catégorielle, et une boîte à moustaches pour comparer la dispersion et repérer les valeurs extrêmes entre plusieurs groupes.

7. Les méthodes d'analyse de données en géographie se regroupent en trois familles : les méthodes descriptives (analyses factorielles, classifications hiérarchiques), les méthodes explicatives (régressions, analyses de variance, modèles linéaires) et les méthodes de prévision, qui reposent sur l'étude des séries chronologiques. Chacune de ces approches répond à une finalité distincte : comprendre, expliquer ou anticiper les phénomènes géographiques.

8. Au cœur de cette étude, des termes clés utilisés en géographie méritent d'être clarifiés :
- La **population statistique** correspond à l'ensemble des unités sur lesquelles porte l'étude, comme l'ensemble des habitants d'un territoire.
- L'**individu statistique** est l'unité élémentaire de cette population ; lorsqu'il est localisable dans l'espace (une ville, une commune, un salarié…), on parle alors d'**unité spatiale**.
- Les **caractères statistiques** sont les propriétés mesurées sur chaque individu (âge, revenu, altitude, catégorie socio-professionnelle,...).
- Les **modalités** représentent les différentes valeurs possibles que peut prendre un caractère ; elles doivent être à la fois exclusives (un individu ne peut appartenir qu'à une seule modalité) et exhaustives (toutes les possibilités doivent être couvertes).

On distingue plusieurs **types de caractères** :

- **Qualitatifs** :
  - Qualitatif nominal, lorsque les catégories n'ont pas d'ordre (ex. : type de logement)
  - *Qualitatif ordinal*, lorsque les catégories sont ordonnées (ex. : niveau d'étude).
- **Quantitatifs** :
  - *Quantitatifs discrets*, lorsqu'ils prennent des valeurs entières (ex. : nombre d'enfants)
  - *Quantitatif continu*, lorsqu'ils varient dans un intervalle (ex. : âge, salaire). Parmi eux, on différencie les variables d'intervalle et de rapport selon la signification du zéro.

Il n'existe pas de hiérarchie en termes de valeur entre ces types de caractères. En revanche, dans une perspective spatiale, on distingue les **unités primaires** (non agrégées, considérées comme des « atomes ») et les **unités secondaires** (agrégées, assimilées à des « molécules »). Cette distinction constitue une forme de hiérarchie utile pour l'analyse spatiale et les processus d'agrégation.

9. En statistique, lorsqu'on regroupe les données d'une variable quantitative en classes, on cherche à caractériser chacune d'elles par deux éléments fondamentaux : l'amplitude et la densité. L'amplitude, notée A, correspond à la largeur de la classe, c'est-à-dire à la différence entre la valeur maximale (notée b) et la valeur minimale (notée a) de cette classe. Elle se calcule par la formule suivante : A = b – a. L'amplitude mesure donc l'étendue locale des valeurs observées dans cette classe. Ensuite, la densité (notée) permet de comparer des classes d'amplitudes différentes. Elle exprime la concentration des individus (ou effectifs) dans chaque classe. Elle se calcule par la formule : où est l'effectif de la classe, et son amplitude. Autrement dit, la densité indique le nombre d'individus par unité d'amplitude.

10. Les formules de Sturges et de Yule servent à déterminer le nombre optimal de classes lorsqu'on construit un tableau ou un histogramme à partir d'une série de données quantitatives. Leur utilisation vise à éviter : un découpage trop fin (trop de classes, entraînant une perte de lisibilité) ou au contraire un découpage trop grossier (trop peu de classes, entraînant une perte d'information). Ces formules permettent donc de trouver un bon équilibre entre précision et clarté, sachant que celle de Yule donne un résultat proche de celui de Sturges, mais s'applique mieux à certains jeux de données.

11. Un effectif représente le nombre d'occurrences d'une modalité donnée, tandis que la fréquence est le rapport entre l'effectif d'une modalité et l'effectif total. La fréquence cumulée s'obtient en additionnant les fréquences successives jusqu'à une modalité donnée. L'ensemble de ces valeurs constitue une distribution statistique, qui décrit la répartition d'un caractère au sein d'une population. Elle permet de dégager des tendances générales et de poser les bases d'une interprétation géographique solide.

II- Mise en œuvre avec Python

Questions 1 à 4 : Après avoir téléchargé les fichiers de la séance 02 et s'être situé dans le dossier de la séance, j'ai ouvert le fichier main.py pour me trouver face à un code recensant les résultats des élections présidentielles du 1er tour de 2022, téléchargé dans le dossier data.

Question 5 : Le fichier CSV est lu à l'aide de l'instruction "with" et de la méthode "read_csv()" de la bibliothèque Pandas. Le contenu est stocké dans une variable contenu puis converti en DataFrame afin d'être affiché dans le terminal.

NotePad++ Questions 5 à 9 :

```
# Question 5
data = pd.DataFrame(contenu)
print(data.head)
# Question 6
print ("nombre de lignes: ",len(contenu))
print ("nombre de colonnes: ",len(contenu.columns))
# Question 7
print(contenu.dtypes)
# Questions 8 et 9
print ("nombre d'inscrits: ",format(contenu["Inscrits"].sum(),",").replace(","," "))
```

Résultats Python **Questions 1 à 9** :

```
PS D:\Master 1 - GAED SCT\Blieck-2025-2026-Analyse-de-donnees\Seance-02\src> python main.py
<bound method NDFrame.head of     Code du dÃ©partement          LibellÃ© du dÃ©partement  Inscrits  ...         Nom.11
 PrÃ©nom.11  Voix.11
0                     01                                Ain    438109  ...  DUPONT-AIGNAN     Nicolas   8998.0
1                     02                              Aisne    373544  ...  DUPONT-AIGNAN     Nicolas   5790.0
2                     03                             Allier    249991  ...  DUPONT-AIGNAN     Nicolas   4216.0
3                     04       Alpes-de-Haute-Provence      128075  ...  DUPONT-AIGNAN     Nicolas   2504.0
4                     05                   Hautes-Alpes     113519  ...  DUPONT-AIGNAN     Nicolas   2142.0
..                   ...                            ...        ...  ...            ...         ...      ...
102                   ZP            PolynÃ©sie franÃ§aise    205576  ...  DUPONT-AIGNAN     Nicolas   1969.0
103                   ZS         Saint-Pierre-et-Miquelon      5045  ...  DUPONT-AIGNAN     Nicolas     82.0
104                   ZW                  Wallis et Futuna      9528  ...  DUPONT-AIGNAN     Nicolas    244.0
105                   ZX    Saint-Martin/Saint-BarthÃ©lemy     24414  ...  DUPONT-AIGNAN     Nicolas    339.0
106                   ZZ  FranÃ§ais Ã©tablis hors de France   1435746  ...  DUPONT-AIGNAN     Nicolas   7074.0

[107 rows x 56 columns]>
nombre de lignes:  107
nombre de colonnes:  56
nombre d'inscrits:  48 747 876
```

Question 7 dans Python : La nature statistique des variables est identifiée à l'aide de l'attribut "dtypes", ce qui permet de distinguer les variables quantitatives des variables qualitatives.

```
[107 rows x 56 columns]>
nombre de lignes:  107
nombre de colonnes:  56
Code du dÃ©partement         object
LibellÃ© du dÃ©partement     object
Inscrits                     int64
Abstentions                float64
Votants                    float64
Blancs                     float64
Nuls                       float64
ExprimÃ©s                  float64
Sexe                        object
Nom                         object
PrÃ©nom                     object
Voix                       float64
Sexe.1                      object
Nom.1                       object
PrÃ©nom.1                   object
Voix.1                     float64
Sexe.2                      object
Nom.2                       object
PrÃ©nom.2                   object
Voix.2                     float64
Sexe.3                      object
Nom.3                       object
PrÃ©nom.3                   object
Voix.3                     float64
Sexe.4                      object
Nom.4                       object
PrÃ©nom.4                   object
Voix.4                     float64
Sexe.5                      object
Nom.5                       object
PrÃ©nom.5                   object
Voix.5                     float64
Sexe.6                      object
Nom.6                       object
PrÃ©nom.6                   object
Voix.6                     float64
Sexe.7                      object
Nom.7                       object
PrÃ©nom.7                   object
Voix.7                     float64
Sexe.8                      object
Nom.8                       object
PrÃ©nom.8                   object
Voix.8                     float64
Sexe.9                      object
Nom.9                       object
PrÃ©nom.9                   object
Voix.9                     float64
Sexe.10                     object
Nom.10                      object
PrÃ©nom.10                  object
Voix.10                    float64
Sexe.11                     object
Nom.11                      object
PrÃ©nom.11                  object
Voix.11                    float64
dtype: object
```

Question 10 sur Notepad++ : Les effectifs sont calculés uniquement pour les colonnes quantitatives, grâce à une condition sur le type des variables. Les résultats sont stockés dans une liste.

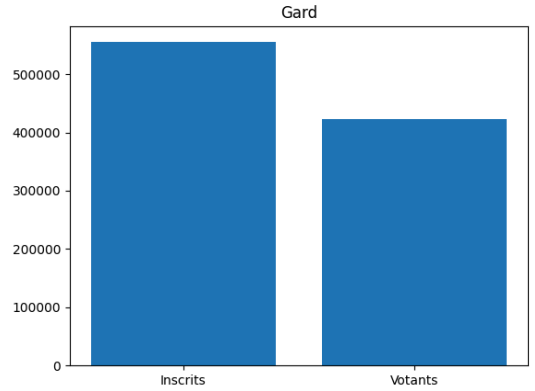
```python
# Question 10
effectifs =[]
for colonne in contenu:
    if contenu.dtypes [colonne] !=object:
        effectifs.append (int(contenu [colonne].sum()))
print (effectifs)
```

Question 10 sur Python :

```
[48747876, 12824169, 35923707, 543609, 247151, 35132947, 197094, 802422, 9783058, 1101387, 8133828, 2485226, 7712520, 61
6478, 1627853, 1679001, 268904, 725176]
PS D:\Master 1 - GAED SCT\Blieck-2025-2026-Analyse-de-donnees\Seance-02\src>
```

Question 11 : Des diagrammes en barres sont générés pour chaque département afin de comparer le nombre d'inscrits et de votants grâce à l'usage de boucles "for". Les images sont enregistrées dans un sous-dossier dédié.

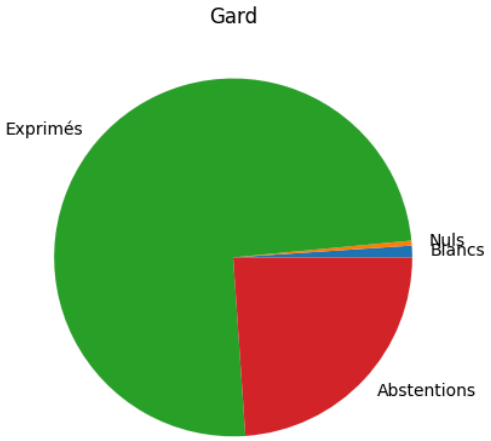
```python
# Question 11
inscrit =[]
for departement in contenu ["Inscrits"]:
    inscrit.append (departement)

votant=[]
for departement in contenu ["Votants"]:
    votant.append (departement)

nom=[]
for departement in contenu ["LibellÃ© du dÃ©partement"]:
    nom.append (departement)


for ins,vot,n in zip(inscrit,votant,nom):
  print (ins,vot,n)
  plt.figure()
  a=np.array (["Inscrits","Votants"])
  b=np.array ([ins,vot])
  plt.bar (a,b)
  plt.title (n)
  plt.savefig ("diagrammes en barres/"+n.replace("/"," ")+".png")
  plt.close()
```



Question 12 : De la même manière, j'ai créé des boucles pour faire des diagrammes circulaires avec les votes blancs, nuls, exprimés et l'abstention pour chaque département. La quatrième boucle permet de créer chaque diagramme dans le dossier « diagrammes circulaires » de mon ordinateur. J'ai rencontré quelques problèmes : sans la commande « plt.close » les données des différents départements étaient additionnées les unes aux autres a lieu d'être réparties par diagramme.

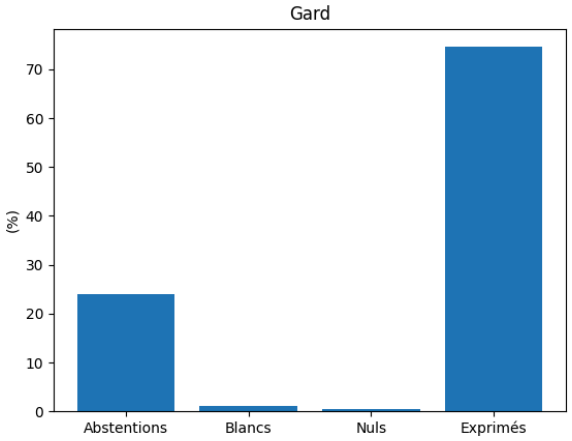
```
#Question 12
blanc=[]
for departement in contenu ["Blancs"]:
    blanc.append (departement)

nul=[]
for departement in contenu ["Nuls"]:
    nul.append (departement)

exprime=[]
for departement in contenu ["ExprimÃ©s"]:
    exprime.append (departement)

abstention=[]
for departement in contenu ["Abstentions"]:
    abstention.append (departement)

for b,nu,e,a,n in zip(blanc,nul,exprime,abstention,nom):
    plt.figure()
    valeur=[b,nu,e,a]
    label=["Blancs","Nuls","Exprimés","Abstentions"]
    plt.pie (valeur,labels=label)
    plt.title (n)
    plt.savefig ("diagrammes circulaire/"+n.replace("/"," ")+".png")
    plt.close()
```

Voici une exemple de diagramme circulaire obtenu :



Question 13 : J'ai utilisé le même procédé que pour les questions précédentes tout en faisant le choix d'utiliser des pourcentages pour faire des histogrammes de la distribution des inscrits. Voici le programme utilisé et un exemple d'histogramme obtenu.

```
#Question 13
for b,nu,e,a,n,ins in zip(blanc,nul,exprime,abstention,nom,inscrit):
    plt.figure()
    y=np.array([(a/ins)*100,(b/ins)*100,(nu/ins)*100,(e/ins)*100])
    x=np.array(["Abstentions","Blancs","Nuls","Exprimés"])
    plt.bar (x,y)
    plt.title (n)
    plt.ylabel("(%)")
    plt.savefig ("histogrammes/"+n.replace("/"," ")+".png")
    plt.close()
```

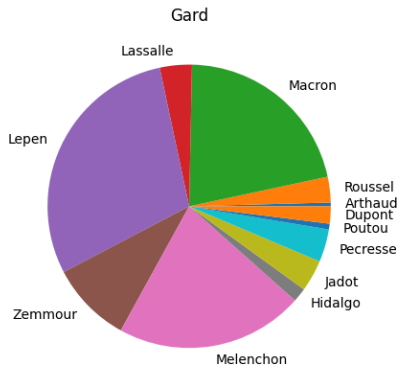


Questions bonus : Je n'ai pas réussi à simplifier davantage le code de cette question bonus mais j'ai réemployé des boucles pour réaliser des diagrammes circulaires pour visualiser, pour chaque département, la répartition des voix par candidat. Puis je réalise une formule pour créer un diagramme circulaire global pour l'ensemble de la France.

```python
#Question bonus

Arthaud=[]
for departement in contenu ["Voix"]:
    Arthaud.append (departement)

Roussel=[]
for departement in contenu ["Voix.1"]:
    Roussel.append (departement)

Macron=[]
for departement in contenu ["Voix.2"]:
    Macron.append (departement)

Lassalle=[]
for departement in contenu ["Voix.3"]:
    Lassalle.append (departement)

Lepen=[]
for departement in contenu ["Voix.4"]:
    Lepen.append (departement)

Zemmour=[]
for departement in contenu ["Voix.5"]:
    Zemmour.append (departement)

Melenchon=[]
for departement in contenu ["Voix.6"]:
    Melenchon.append (departement)

Hidalgo=[]
for departement in contenu ["Voix.7"]:
    Hidalgo.append (departement)

Jadot=[]
for departement in contenu ["Voix.8"]:
    Jadot.append (departement)

Pecresse=[]
for departement in contenu ["Voix.9"]:
    Pecresse.append (departement)

Poutou=[]
for departement in contenu ["Voix.10"]:
    Poutou.append (departement)

Dupont=[]
for departement in contenu ["Voix.11"]:
    Dupont.append (departement)
```

```python
for (arthaud,roussel,macron,lassalle,lepen,zemmour,melenchon,hidalgo,jadot,pecresse,poutou,dupont,n) in zip(Arthaud,Roussel,Macron,Lassalle,Lepen,Zemmour,Melenchon,Hidalgo,Jadot,Pecresse,Poutou,Dupont,no
    plt.figure()
    a=[arthaud,roussel,macron,lassalle,lepen,zemmour,melenchon,hidalgo,jadot,pecresse,poutou,dupont]
    b=["Arthaud","Roussel","Macron","Lassalle","Lepen","Zemmour","Melenchon","Hidalgo","Jadot","Pecresse","Poutou","Dupont"]
    plt.pie (a,labels=b)
    plt.title (n)
    plt.savefig ("bonus/"+n.replace("/"," ")+".png")
    plt.close()

plt.figure()
a=[sum(Arthaud),sum(Roussel),sum(Macron),sum(Lassalle),sum(Lepen),sum(Zemmour),sum(Melenchon),sum(Hidalgo),sum(Jadot),sum(Pecresse),sum(Poutou),sum(Dupont)]
b=["Arthaud","Roussel","Macron","Lassalle","Lepen","Zemmour","Melenchon","Hidalgo","Jadot","Pecresse","Poutou","Dupont"]
plt.pie(a,labels=b)
plt.title("Total France")
plt.savefig ("bonus/Total.png")
plt.close()
```

Voici un exemple de diagramme obtenu :



Séance 3

I- Questions de cours

1. Le caractère le plus général est le caractère qualitatif, car il permet de décrire une réalité sous forme de catégories (comme le sexe, la couleur, la profession ou le type de logement). Le caractère quantitatif n'est qu'un cas particulier du qualitatif, puisqu'il correspond uniquement aux situations où ces catégories peuvent être exprimées par des nombres. Or, tous les phénomènes ne sont pas mesurables numériquement. C'est pour cette raison que le caractère qualitatif est plus général que le caractère quantitatif.

2. Distinguer les caractères quantitatifs discrets et continus est essentiel pour garantir une analyse statistique correcte et une interprétation précise des données. Les caractères quantitatifs discrets permettent de compter et de regrouper facilement les valeurs, ce qui simplifie les calculs statistiques.
À l'inverse, les caractères quantitatifs continus offrent une plus grande précision dans la mesure des phénomènes, car ils peuvent prendre une infinité de valeurs, mais ils nécessitent des méthodes d'analyse spécifiques pour intégrer les valeurs intermédiaires. Cette distinction est donc fondamentale dans de nombreux domaines.

3. a) Il existe plusieurs types de moyennes afin de répondre à des objectifs différents dans l'analyse des données. Chaque moyenne a un usage précis. La moyenne arithmétique est, par exemple, la plus utilisée pour les variables quantitatives. Il existe d'autres types de moyennes tels que les moyennes géométriques pour calculer le taux croissance, les moyennes quadratiques pour les phénomènes physiques ou les moyennes mobiles pour l'analyse des séries chronologiques. Chaque type de moyenne est donc adapté à un contexte particulier et permet d'obtenir des valeurs discrètes et/ou continues.

b) On calcule la médiane pour déterminer la valeur centrale d'une série statistique ordonnée, c'est-à-dire la valeur qui partage les données en deux parties égales : 50 % des valeurs sont inférieures, 50 % des valeurs sont supérieures. Elle est d'ailleurs appelée la « moyenne du milieu ». Elle constitue donc un indicateur essentiel de la tendance centrale d'une distribution. L'un des principaux avantages de la médiane est qu'elle est résistante aux valeurs extrêmes (valeurs aberrantes). Contrairement à la moyenne, elle n'est pas influencée par des valeurs très grandes ou très petites. La médiane est également particulièrement adaptée aux distributions asymétriques. Dans ce cas, elle fournit une meilleure indication de la tendance centrale que la moyenne. Sur le plan pratique, la médiane est très utilisée dans de nombreux domaines, comme l'économie (niveau de vie, revenus), car elle permet de prendre des décisions basées sur une mesure fiable et représentative de la population étudiée. Elle offre ainsi une vision plus juste et plus précise des données, indispensable pour l'analyse statistique et la prise de décision.

c) On calcule le mode pour identifier la valeur la plus fréquente d'une série statistique. Le mode, aussi appelé valeur dominante, correspond à la modalité ayant l'effectif maximal (pour une variable discrète) ou à la plus forte densité de probabilité (pour une variable continue). Il s'agit donc d'une moyenne de fréquence, qui indique ce qui apparaît le plus souvent dans les données. Le mode est un outil essentiel pour l'analyse de la distribution,

car il permet de repérer les valeurs dominantes, mettre en évidence des tendances générales mais aussi identifier des comportements typiques dans une population. Cependant, le mode n'existe pas toujours. Lorsqu'il existe, il peut être unique ou multiple : on parle alors de distribution bimodale ou plurimodale. Lorsque plusieurs modes apparaissent après un regroupement en classes, cela peut indiquer la présence de plusieurs populations différentes, chacune ayant ses propres caractéristiques. Dans ce cas, la moyenne arithmétique n'est plus un bon indicateur de tendance centrale. Le mode permet donc d'identifier la valeur la plus représentative d'un point de vue pratique et comportemental, et il est particulièrement utile pour analyser les préférences, les répétitions et les dominances dans une distribution statistique.

4. La médiale est la valeur centrale qui partage la masse totale des valeurs en deux parts égales. Elle permet de connaître le centre réel de la distribution et de détecter la concentration des valeurs autour du centre. L'indice de Gini mesure l'inégalité globale d'une distribution, variant de 0 (égalité parfaite) à 1 (inégalité totale). Il synthétise la répartition des valeurs dans la population et permet de comparer la concentration ou les inégalités entre populations ou périodes. Ainsi, la médiale fournit une mesure centrale, tandis que le Gini donne une mesure globale de concentration, formant un outil complet pour analyser les inégalités économiques ou sociales.

5. a) On calcule la variance à la place des écarts à la moyenne car la somme des écarts simples est toujours égale à zéro : les valeurs positives et négatives se compensent, ce qui empêche de mesurer la dispersion. En élevant les écarts au carré, on évite cette compensation et on obtient une mesure correcte de la dispersion autour de la moyenne : la variance. Elle permet ainsi de quantifier précisément l'étalement des données. On remplace la variance par l'écart type car la variance est exprimée dans une unité au carré, ce qui rend son interprétation peu intuitive. L'écart type étant la racine carrée de la variance, il est exprimé dans la même unité que les données, ce qui permet une interprétation plus simple et plus concrète de la dispersion.

b) On calcule l'étendue car elle permet de mesurer l'amplitude d'une série statistique en faisant la différence entre la valeur maximale et la valeur minimale : $E=x_{max}-x_{min.}$ Elle est facile à calculer, mais comme elle ne dépend que des valeurs extrêmes et ne tient compte ni du nombre de données ni des valeurs intermédiaires, elle devient peu représentative lorsque l'effectif est important, et est donc peu utilisée au-delà d'une dizaine de données.

c) Un quantile sert à partager une série statistique ordonnée en parts égales afin d'analyser la répartition des valeurs dans une population. Il permet donc de situer une donnée par rapport à l'ensemble des observations. Les quantiles les plus utilisés sont : les quartiles Q1, Q2, Q3 qui partagent la série en quatre parties égales (avec Q2 qui correspond à la médiane), les déciles, qui partagent la série en dix parties et les centiles, qui la partagent en cent parties.

d) La boîte de dispersion (ou boîte à moustaches) permet de représenter graphiquement la distribution d'une série statistique à partir des quartiles, de la médiane et des valeurs extrêmes. Elle sert à visualiser la dispersion, à repérer l'asymétrie et à comparer plusieurs séries statistiques. La boîte s'étend de Q1 à Q3, ce qui représente 50 % des valeurs, et la médiane est indiquée par un trait. Les moustaches relient la boîte aux valeurs

minimale et maximale. La longueur de la boîte indique la dispersion centrale et la position de la médiane renseigne sur la symétrie de la distribution.

6. a) D'un côté, les moments centrés sont calculés à partir des écarts à la moyenne. Ils servent à décrire les caractéristiques fondamentales d'une distribution. On distingue ainsi le moment centré d'ordre 1 qui est toujours nul par définition, car il correspond à la moyenne des écarts par rapport à la moyenne, du moment centré d'ordre 2 qui correspond à la variance et mesure la dispersion. Le moment centré d'ordre 3 permet d'étudier l'asymétrie tandis que le moment centré d'ordre 4 renseigne sur l'aplatissement de la distribution. De leur côté, les moments absolus, quant à eux, utilisent la valeur absolue des écarts par rapport à un point donné. Ils permettent de mesurer les écarts moyens sans compensation entre valeurs positives et négatives. Ainsi, les moments centrés et les moments absolus sont complémentaires et permettent, ensemble, de décrire complètement la forme d'une distribution statistique.

b) On vérifie la symétrie d'une distribution pour savoir si elle est équilibrée et ressemble à une loi normale. Dans une distribution symétrique, la moyenne, la médiane et le mode sont égaux. La symétrie peut être mesurée avec le coefficient d'asymétrie β1 : il est positif si la distribution est étalée vers la droite, négatif si elle est étalée vers la gauche, et nul si la distribution est parfaitement symétrique. On peut aussi regarder l'aplatissement avec le coefficient β2 : il est positif pour une distribution plus plate que la normale, négatif pour une distribution plus pointue, et nul pour une distribution normale. Ces coefficients sont des estimations, donc pour un échantillon il faut utiliser des formules corrigées pour obtenir des résultats fiables.

II- Mise en œuvre avec Python

Questions 1 à 4 : enregistrement des fichiers, ouverture du dossier main.py dans Notepad++ pour pouvoir débuter la séance.

Questions 5 à 7 : Pour éviter de répéter les mêmes lignes de codes, j'ai condensé les trois questions. Voici donc le code nous permettant de sélectionner les colonnes contenant des caractères quantitatifs et de calculer sous forme de liste les moyennes, médianes, modes, écarts-types, écarts-absolus et étendues pour chaque colonne.

Question 5 : Les colonnes quantitatives ont été sélectionnées en excluant les variables de type object. Pour chacune d'elles, les paramètres statistiques suivants ont été calculés à l'aide des méthodes Pandas appropriées, puis arrondis à deux décimales : moyenne , médiane, mode, écart-type, écart absolu à la moyenne et étendue.

Question 6 : Les listes de paramètres statistiques calculées ont été affichées directement dans le terminal.

Question 7 : La distance interquartile (IQR) et la distance interdécile (IDR) ont été calculées pour chaque colonne quantitative à l'aide de la méthode quantile().

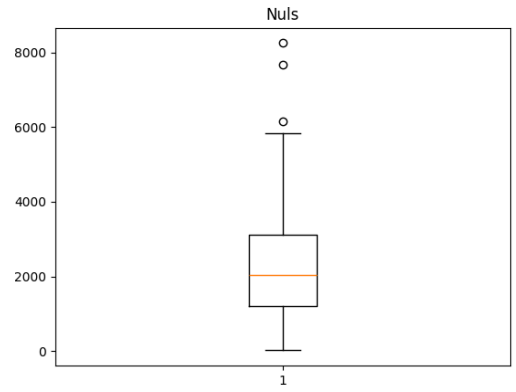
```python
#Questions 5 à 7
Moyenne=[]
Mediane=[]
Mode=[]
Ecarttype=[]
Ecartabsolu=[]
Etendue=[]
IQRs=[]
IDRs=[]
for colonne in contenu:
    if contenu.dtypes [colonne] !=object:
        Moyenne.append(float(contenu[colonne].mean().round(2)))
        Mediane.append(int(contenu[colonne].median()))
        Mode.append(int(contenu[colonne].mode().iloc[0]))
        Ecarttype.append(float(contenu[colonne].std().round(2)))
        Ecartabsolu.append(float((contenu[colonne]-contenu[colonne].mean()).abs().mean().round(2)))
        Etendue.append(int(contenu[colonne].max()-contenu[colonne].min()))
        q1=contenu[colonne].quantile(0.25)
        q3  = contenu[colonne].quantile(0.75)
        d1  = contenu[colonne].quantile(0.10)
        d9  = contenu[colonne].quantile(0.90)
        IQRs.append(float(q3 - q1))
        IDRs.append(float((d9 - d1).round(2)))

print ("Moyenne:\t",Moyenne)
print ("Médiane:\t",Mediane)
print ("Mode:\t",Mode)
print ("Ecart type:\t",Ecarttype)
print ("Ecart absolu:\t",Ecartabsolu)
print ("Etendue:\t",Etendue)
print ("Distance interquartile:\t",IQRs)
print ("Distance interdécile:\t",IDRs)
```

```
Moyenne:         [455587.63, 119852.05, 335735.58, 5080.46, 2309.82, 328345.3, 1842.0, 7499.27, 91430.45, 10293.34, 7601
7.08, 23226.41, 72079.63, 5761.48, 15213.58, 15691.6, 2513.12, 6777.35]
Médiane:         [366859, 95369, 274372, 4001, 2039, 268568, 1627, 5968, 67831, 8944, 64543, 16885, 51556, 4881, 9561, 1
1918, 2118, 6152]
Mode:    [5045, 2272, 2773, 4577, 17, 2701, 1203, 19, 534, 17010, 459, 9657, 501, 75, 72, 51, 3663, 7271]
Ecart type:      [351003.78, 117017.8, 258393.81, 3492.52, 1501.38, 253758.58, 1268.37, 6501.29, 77226.14, 7464.32, 6027
8.1, 20760.6, 66210.68, 4581.79, 14807.62, 13027.13, 1781.41, 4636.02]
Ecart absolu:    [272240.72, 74959.07, 201517.17, 2817.95, 1131.99, 197762.2, 977.36, 4474.96, 59929.14, 5140.37, 42514.
72, 15278.36, 49157.01, 3333.34, 11136.57, 9432.01, 1404.5, 3689.5]
Etendue:         [1808861, 929183, 1297100, 17389, 8236, 1272080, 7651, 45883, 372286, 48168, 372668, 108537, 316871, 22
826, 80196, 69513, 8686, 20535]
Distance interquartile:  [401050.0, 106489.0, 301770.5, 4852.5, 1917.0, 296870.5, 1517.5, 6264.5, 101317.0, 7999.5, 6334
2.0, 20638.5, 60743.5, 4779.0, 14833.5, 13265.5, 2466.0, 6146.5]
Distance interdécile:    [793988.8, 193676.2, 602687.2, 8845.8, 3240.6, 590169.2, 3015.6, 13104.2, 177340.2, 13813.0, 13
0094.6, 43668.8, 159421.2, 10712.2, 38190.8, 27686.8, 4266.6, 12311.0]
```

Question 8 : Des boîtes à moustaches ont été générées pour chaque colonne quantitative à l'aide de Matplotlib puis stockées dans un dossier "boite à moustache".

```python
#Question 8
for colonne in contenu :
    if contenu.dtypes [colonne] !=object:
        plt.figure()
        plt.boxplot(contenu[colonne])
        plt.title(colonne)
        plt.savefig("boite à moustache/"+colonne+".png")
        plt.close()
```

Voici un exemple de boite à moustache obtenu :

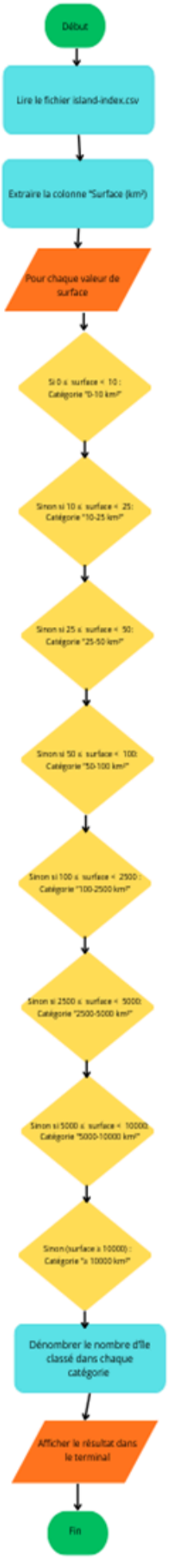


Questions 9 et 10 : J'ai rencontré de grosses difficultés avec le fichier *island-index.csv* car, pour la question 10, il était demandé de sélectionner la colonne « Surface (km²) ». Or, cette colonne n'était pas trouvée par mon algorithme. Après de nombreux essais je crois que mon code continue de se tromper de colonne dans son affichage de données. J'y ais passé un temps fou et je suis incapable d'expliquer comment résoudre ce problème. J'ai donc finalement fait l'exercice avec des données sans doute fausses mais, au moins, j'ai tout de même pu comprendre comment catégoriser et dénombrer des éléments en fonction de leur surface.

```
#Question 10
with open("./data/island-index.csv","r",encoding="latin-1") as fichier:
    doc = pd.read_csv(fichier,low_memory=False)
print (doc["Surface (kmÂ²)"])
limite=[0,10,25,50,100,2500,5000,10000]
categorie=[0,0,0,0,0,0,0,0]
for ile in doc["Surface (kmÂ²)"]:
    for i in range (len(limite)):
        if i==len(limite)-1 or ile <= limite [i+1]:
            categorie [i]+=1
print (categorie)
for i in range (len(categorie)-1):
    print ("Nombre d'îles entre",limite[i],"km² et",limite [i+1],"km²:",categorie[i])
print ("Nombre d'îles supérieur à 10 000 km²: ",categorie [7])
```

```
Name: Surface (kmÂ²), Length: 84219, dtype: float64
[78423, 80750, 81914, 82702, 84048, 84108, 84148, 84219]
Nombre d'îles entre 0 km² et 10 km²: 78423
Nombre d'îles entre 10 km² et 25 km²: 80750
Nombre d'îles entre 25 km² et 50 km²: 81914
Nombre d'îles entre 50 km² et 100 km²: 82702
Nombre d'îles entre 100 km² et 2500 km²: 84048
Nombre d'îles entre 2500 km² et 5000 km²: 84108
Nombre d'îles entre 5000 km² et 10000 km²: 84148
Nombre d'îles supérieur à 10 000 km²:  84219
```

J'ai ensuite tenté de concevoir un organigramme explicitant la solution en employant des formes associées au type d'action réalisé : des ovales pour le début et la fin du processus, des rectangles pour les actions, des parallélogrammes pour les entrées ou sorties et des losanges pour les conditions. Les flèches quant à elles indiquent le sens du déroulement.


Début
Lire le fichier island-index.csv
Extraire la colonne "Surface (km²)
Pour chaque valeur de surface
Si 0 ≤ surface < 10 :
Catégorie "0-10 km²"
Sinon si 10 ≤ surface < 25:
Catégorie "10-25 km²"
Sinon si 25 ≤ surface < 50:
Catégorie "25-50 km²"
Sinon si 50 ≤ surface < 100:
Catégorie "50-100 km²"
Sinon si 100 ≤ surface < 2500 :
Catégorie "100-2500 km²"
Sinon si 2500 ≤ surface < 5000:
Catégorie "2500-5000 km²"
Sinon si 5000 ≤ surface < 10000:
Catégorie "5000-10000 km²"
Sinon (surface ≥ 10000) :
Catégorie "≥ 10000 km²"
Dénombrer le nombre d'île classé dans chaque catégorie
Afficher le résultat dans le terminal
Fin

La taille importante de mon organigramme l'empêchant d'être totalement lisible, veuillez trouver ci-joint le contenu des différentes cases :

- Début
- Lire le fichier island-index.csv
- Extraite la colonne « Surface (km²) »
- Pour chaque valeur de surface
- Si 0 ≤ surface < 10 : Catégorie "0-10 km²"
- Sinon si 10 ≤ surface < 25 : Catégorie "10-25 km²"
- Sinon si 25 ≤ surface < 50 : Catégorie "25-50 km²"
- Sinon si 50 ≤ surface < 100 : Catégorie "50-100 km²"
- Sinon si 100 ≤ surface < 2500 : Catégorie "100-2500 km²"
- Sinon si 2500 ≤ surface < 5000 : Catégorie "2500-5000 km²"
- Sinon si 5000 ≤ surface < 10000 : Catégorie "5000-10000 km²"
- Sinon (surface ≥ 10000) : Catégorie "≥ 10000 km²"
- Dénombrer le nom d'îles classées dans chaque catégorie
- Afficher le résultat dans le terminal
- Fin

Séance 4

I- Question de cours

1. Plusieurs critères peuvent être mis en avant pour choisir entre une distribution à variables discrètes et une distribution à variables continues.

Le premier critère déterminant dans le choix d'une distribution statistique est la nature intrinsèque de la variable étudiée. Lorsqu'un phénomène produit des valeurs dénombrables, souvent issues d'un comptage, par exemple le nombre de naissances, d'habitants ou d'événements observés dans un espace donné, la variable est discrète. Les distributions associées, telles que celles issues des épreuves de Bernoulli, la loi binomiale ou encore la loi de Poisson, sont conçues pour représenter des situations où l'on observe des occurrences distinctes, parfois rares, et où l'on peut relier la probabilité d'un événement à la structure de l'énumération. À l'inverse, lorsqu'on analyse une grandeur pouvant prendre toute valeur dans un intervalle, comme un revenu, une altitude, une distance ou une température, la variable est continue. On recourt alors à des lois à densité (normale, log-normale, exponentielle ou encore triangulaire) capables de décrire une répartition fluide et non discrète de la probabilité.

Un second critère repose sur la forme empirique de la distribution observée. Une distribution en « marches », caractéristique d'une fonction de répartition discontinue, signale une variable discrète ; à contrario, une courbe régulière et continue renvoie à une densité de probabilité associée à une variable continue. L'analyse graphique joue ici un rôle central pour guider la sélection.

Enfin, le choix du modèle statistique dépend aussi étroitement du processus génératif du phénomène. Certains mécanismes produisent naturellement des formes particulières : l'agrégation d'un grand nombre d'effets indépendants tend vers la loi normale, les processus multiplicatifs (croissance proportionnelle, effets cumulatifs) génèrent des distributions log-normales ; les comptages d'événements rares dans un espace ou un intervalle de temps se modélisent généralement par une loi de Poisson… L'échelle d'observation et la précision des données influencent peuvent également influencer notre choix : une variable continue arrondie peut prendre l'apparence d'une variable discrète, ce qui impose un choix prudent et réfléchi du modèle.

2. En géographie, certaines lois statistiques sont tout particulièrement mobilisées comme modèles permettant d'interpréter la répartition et les dynamiques des phénomènes spatiaux. Plusieurs distributions revêtent une importance particulière en raison de leur adéquation avec les dynamiques territoriales :

- La **loi de Zipf-Mandelbrot**, ou loi rang-taille, occupe une place centrale dans l'étude des systèmes urbains. En reliant le rang d'une unité urbaine à sa taille, elle met en évidence la hiérarchie et les rapports de domination au sein d'un réseau urbain, révélant la structure inégalitaire qui caractérise de nombreux systèmes territoriaux. Son importance tient au fait qu'elle offre un pont direct entre une distribution statistique et une théorie géographique de l'organisation de l'espace.

- La **loi de Poisson** est fréquemment employée pour modéliser la distribution spatiale d'événements rares et indépendants : points d'accidents, localisation de commerces ou d'équipements, occurrences biologiques... Elle permet de mettre en évidence les surdensités ou déficits locaux par comparaison à une distribution « au hasard », constituant un outil essentiel pour l'analyse des structures ponctuelles.

- La **loi normale**, malgré les réserves historiques quant à sa généralité, reste largement mobilisée dès lors que l'on s'intéresse à des variations issues d'une pluralité de facteurs indépendants : erreurs de mesure, fluctuations naturelles, diffusion de phénomènes à effets marginaux multiples. Elle joue alors un rôle fondamental pour l'inférence statistique, la construction d'intervalles de confiance ou la modélisation de phénomènes centrés autour d'une moyenne.

- La **loi log-normale**, associée à la croissance proportionnelle (effet de Gibrat), décrit des phénomènes où les variations successives s'accumulent de manière multiplicative. Elle s'applique à de nombreuses grandeurs géographiques : taille des villes, distribution des revenus, surfaces de parcelles, volumes de flux... Son usage est justifié par le fait que de nombreux processus spatiaux relèvent d'une dynamique cumulative ou hiérarchique.

- La **loi de Pareto**, proche conceptuellement des distributions à queue lourde, permet de caractériser des organisations spatiales où une petite proportion d'entités concentre une part très importante de la ressource : grandes villes, grandes exploitations agricoles, valeurs immobilières. Elle met en lumière les phénomènes d'inégalité et de concentration qui structurent fortement les territoires.

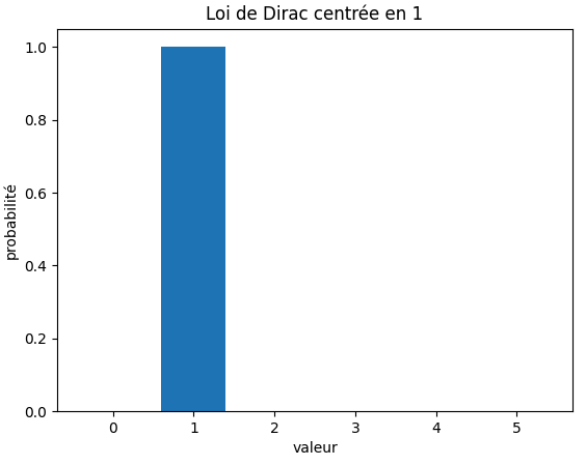
En somme, ces lois sont privilégiées en géographie parce qu'elles permettent de traduire statistiquement des mécanismes territoriaux fondamentaux (hiérarchie, répartition, concentration, hasard spatial) et d'articuler la forme observée des phénomènes à leur logique de production.

II- Mise en œuvre Python

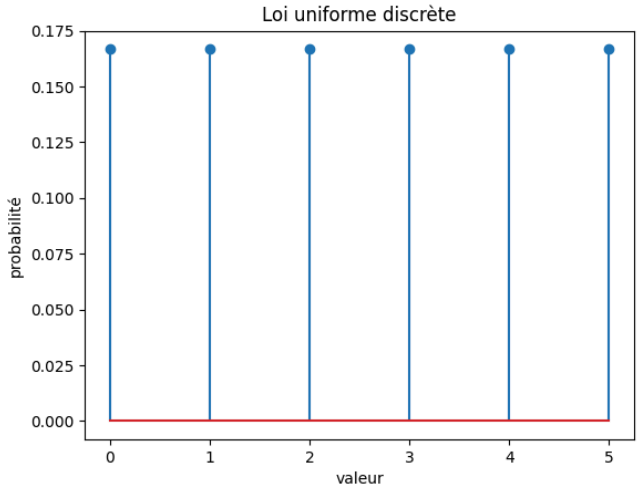
Questions 1 et 2 : N'étant pas sûre des attendus cachés derrière la consigne « visualiser », j'ai pris le parti de le faire sous la forme de graphiques. J'ai donc commencé par importer Matplotlib puis j'ai réutilisé les connaissances acquises lors de la séance 2. J'ai donc fait un code par loi pour que chacune ait un diagramme représentant son fonctionnement. J'en ai profité pour également calculer la moyenne et l'écart type des distributions.

→ **Variables directes :**

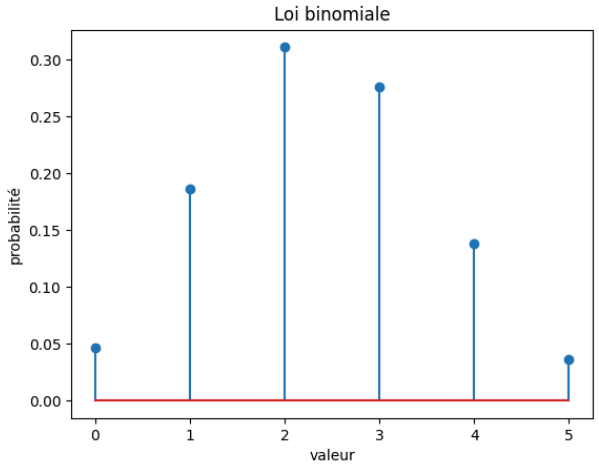
```
#Loi de Dirac
centre=1
x=[0,1,2,3,4,5]
Dirac=scipy.stats.rv_discrete(values=([centre],[1]))
y=Dirac.pmf(x)
plt.figure()
plt.title ("Loi de Dirac centrée en 1")
plt.bar(x,y)
plt.xlabel ("valeur")
plt.ylabel ("probabilité")
plt.show()
plt.close()
print ("DIRAC Moyenne:",Dirac.mean(),"Ecart type:",Dirac.std())
```



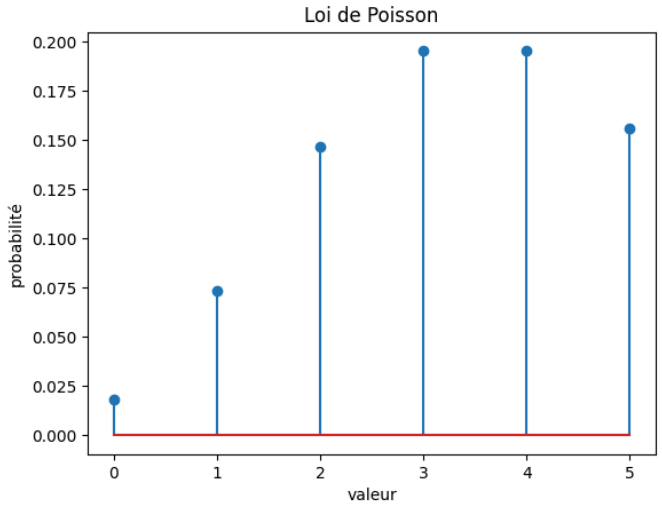

```
#Loi uniforme discrète
distribution=scipy.stats.randint(0,6)
plt.figure()
plt.stem(x,distribution.pmf(x))
plt.title ("Loi uniforme discrète")
plt.xlabel ("valeur")
plt.ylabel ("probabilité")
plt.show()
plt.close()
print ("UNIFORME DISCRETE Moyenne:",distribution.mean(),"Ecart type:",distribution.std())
```



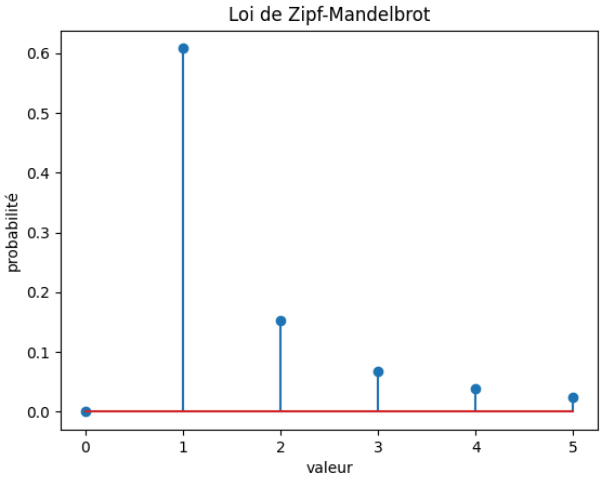
```
#Loi binomiale
distribution=scipy.stats.binom(6,0.4)
plt.figure()
plt.stem(x,distribution.pmf(x))
plt.title ("Loi binomiale")
plt.xlabel("valeur")
plt.ylabel("probabilité")
plt.show()
plt.close()
print ("BINOMIALE Moyenne:",distribution.mean(),"Ecart type:",distribution.std())
```




```
#Loi de Poisson
distribution=scipy.stats.poisson(4)
plt.figure()
plt.stem(x,distribution.pmf(x))
plt.title ("Loi de Poisson")
plt.xlabel("valeur")
plt.ylabel("probabilité")
plt.show()
plt.close()
print ("POISSON Moyenne:",distribution.mean(),"Ecart type:",distribution.std())
```

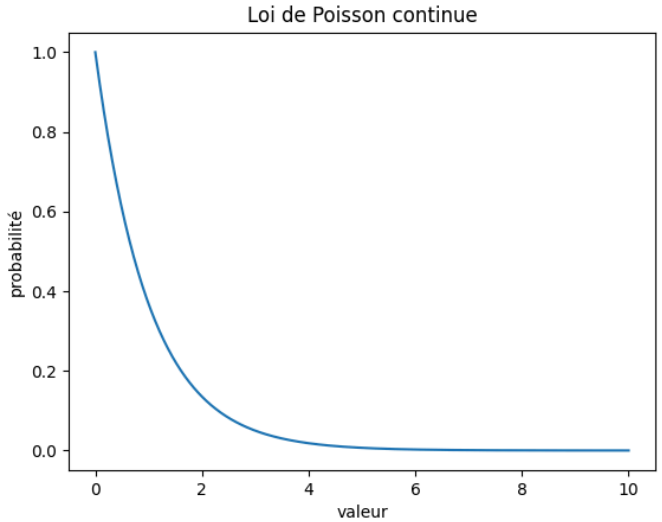


```
#Loi de Zipf-Mandelbrot
distribution=scipy.stats.zipf(2)
plt.figure()
plt.stem(x,distribution.pmf(x))
plt.title ("Loi de Zipf-Mandelbrot")
plt.xlabel("valeur")
plt.ylabel("probabilité")
plt.show()
plt.close()
print ("ZIPF-MANDELBROT Moyenne:",distribution.mean(),"Ecart type:",distribution.std())
```

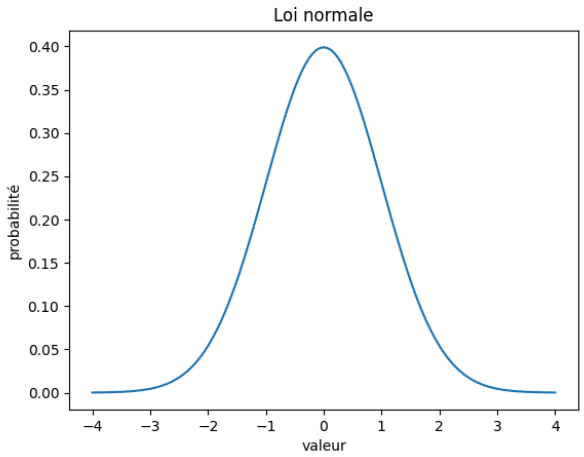


→ **Variables continues :**

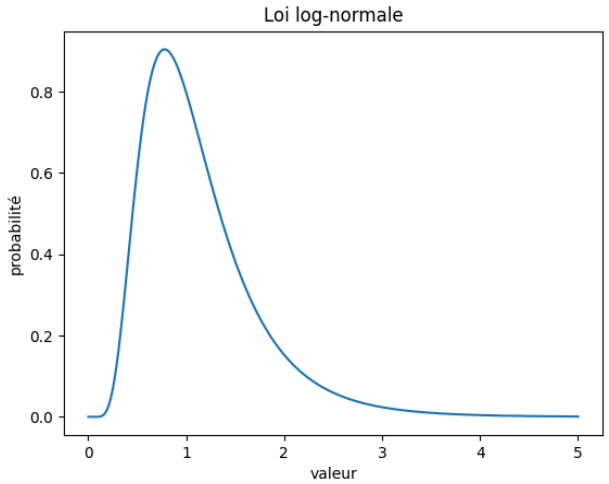
```
#Loi de Poisson continue (exponentielle)
x=np.linspace(0,10,500)
p=scipy.stats.expon.pdf(x)
plt.figure()
plt.plot(x,p)
plt.title("Loi de Poisson continue")
plt.xlabel("valeur")
plt.ylabel("probabilité")
plt.show()
plt.close()
dist=scipy.stats.expon(scale=1/3)
print ("POISSON CONTINUE Moyenne:",dist.mean(),"Ecart type:",dist.std())
```



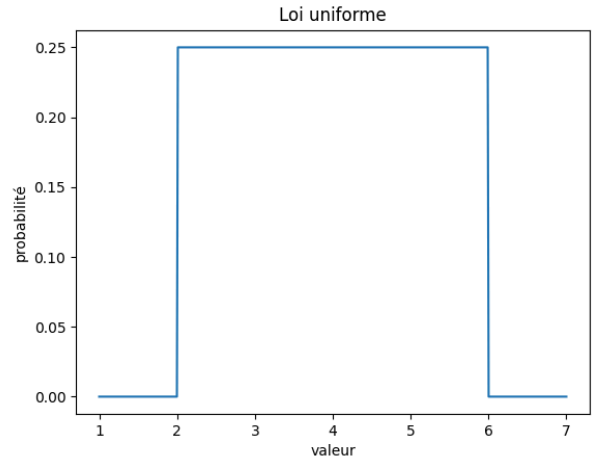
```
#Loi normale
moyenne=0
ecart=1
x=np.linspace(moyenne-4*ecart,moyenne+4*ecart,500)
p=scipy.stats.norm.pdf(x,loc=moyenne,scale=ecart)
plt.figure()
plt.plot(x,p,label=f"N({moyenne},{ecart**2})")
plt.title("Loi normale")
plt.xlabel("valeur")
plt.ylabel("probabilité")
plt.show()
plt.close()
dist=scipy.stats.norm(loc=moyenne,scale=ecart)
print ("NORMALE Moyenne:",dist.mean(),"Ecart type:",dist.std())
```



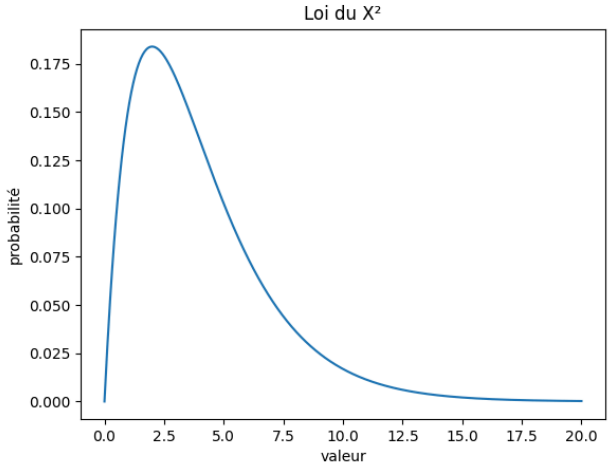

```
#Loi log-normale
moyenne=0
ecart=0.5
x=np.linspace(0.001,5,500)
p=scipy.stats.lognorm(s=ecart,scale=np.exp(moyenne)).pdf(x)
plt.figure()
plt.plot(x,p,label=f"LogNorm(mu={moyenne},sigma={ecart})")
plt.title("Loi log-normale")
plt.xlabel("valeur")
plt.ylabel("probabilité")
plt.show()
plt.close()
dist=scipy.stats.lognorm(s=ecart,scale=np.exp(moyenne))
print ("LOGNORMALE Moyenne:",dist.mean(),"Ecart type:",dist.std())
```



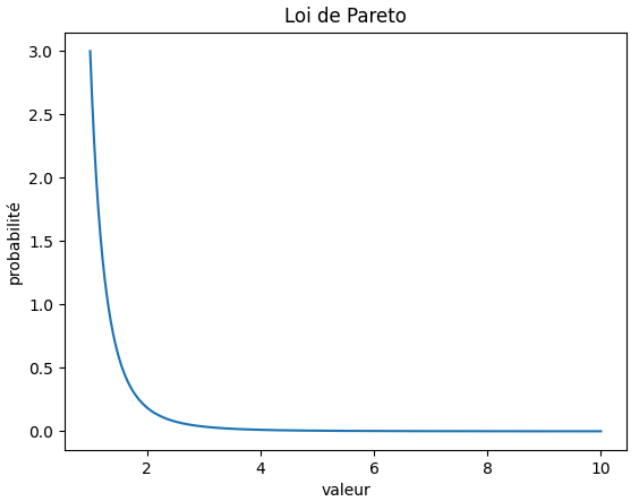
```
#Loi uniforme
inf=2
sup=6
x=np.linspace(inf-1,sup+1,500)
p=scipy.stats.uniform(loc=inf,scale=sup-inf).pdf(x)
plt.figure()
plt.plot(x,p,label=f"Uniforme({inf},{sup})")
plt.title("Loi uniforme")
plt.xlabel("valeur")
plt.ylabel("probabilité")
plt.show()
plt.close()
dist=scipy.stats.uniform(loc=inf,scale=sup-inf)
print ("UNIFORME Moyenne:",dist.mean(),"Ecart type:",dist.std())
```




```
#Loi du X²
k=4
x=np.linspace(0,20,500)
p=scipy.stats.chi2(df=k).pdf(x)
plt.figure()
plt.plot(x,p,label=f"Chi²(k={k})")
plt.title("Loi du X²")
plt.xlabel("valeur")
plt.ylabel("probabilité")
plt.show()
plt.close()
dist=scipy.stats.chi2(df=k)
print ("X² Moyenne:",dist.mean(),"Ecart type:",dist.std())
```



```python
#Loi de Pareto
forme=3
scale=1
x=np.linspace(scale,10,500)
p=scipy.stats.pareto(b=forme,scale=scale).pdf(x)
plt.figure()
plt.plot(x,p,label=f"Pareto(alpha={forme},xm={scale})")
plt.title("Loi de Pareto")
plt.xlabel("valeur")
plt.ylabel("probabilité")
plt.show()
plt.close()
dist=scipy.stats.pareto(b=forme,scale=scale)
print ("PARETO Moyenne:",dist.mean(),"Ecart type:",dist.std())
```



Et voici les moyennes et écarts types obtenus pour chaque variable :

```
DIRAC Moyenne: 1.0 Ecart type: 0.0
UNIFORME DISCRETE Moyenne: 2.5 Ecart type: 1.707825127659933
BINOMIALE Moyenne: 2.4000000000000004 Ecart type: 1.2000000000000002
POISSON Moyenne: 4.0 Ecart type: 2.0
ZIPF-MANDELBROT Moyenne: inf Ecart type: inf
POISSON CONTINUE Moyenne: 0.3333333333333333 Ecart type: 0.3333333333333333
NORMALE Moyenne: 0.0 Ecart type: 1.0
LOGNORMALE Moyenne: 1.1331484530668263 Ecart type: 0.6039005332108811
UNIFORME Moyenne: 4.0 Ecart type: 1.1547005383792515
X² Moyenne: 4.0 Ecart type: 2.8284271247461903
PARETO Moyenne: 1.5 Ecart type: 0.8660254037844386
```

Séance 5

I- Questions de cours

1. L'échantillonnage désigne l'opération qui consiste à prélever une partie d'une population statistique, appelée population mère, afin d'en étudier les caractéristiques et d'en inférer les paramètres globaux. En statistique inférentielle, cette démarche est fondamentale car l'étude exhaustive d'une population est, dans la majorité des cas, soit impossible, soit trop coûteuse en temps, en moyens financiers ou en ressources humaines. Les méthodes d'échantillonnage se divisent en deux grandes catégories : les méthodes aléatoires et les méthodes non aléatoires. Les méthodes aléatoires reposent sur un tirage au sort, garantissant l'équiprobabilité des individus. Les méthodes non aléatoires, quant à elles, reposent sur une construction raisonnée de l'échantillon, comme l'échantillonnage systématique (sélection à intervalles réguliers) ou la méthode des quotas, qui cherche à respecter les proportions connues de certaines caractéristiques dans la population. Le choix d'une méthode d'échantillonnage dépend principalement de l'existence d'une base de sondage, du coût de collecte, de la précision attendue et du risque de biais. Un principe essentiel guide ce choix : un petit échantillon représentatif est toujours préférable à un grand échantillon biaisé, car seule la représentativité permet une inférence fiable.

2. Un estimateur est une variable aléatoire construite comme une fonction des observations aléatoires issues d'un échantillon. Il sert à approcher un paramètre inconnu de la population, noté en général θ (par exemple une moyenne, une variance ou une proportion). Formellement, un estimateur s'écrit comme une fonction des variables aléatoires observées. L'estimation, quant à elle, correspond à la valeur numérique concrète obtenue lorsque l'on applique cet estimateur aux données effectivement observées. Autrement dit, l'estimateur appartient au domaine théorique des probabilités, tandis que l'estimation relève du calcul empirique. Ce processus constitue le cœur de la statistique inférentielle : on passe de l'observation à l'évaluation d'un paramètre inconnu.

3. L'intervalle de fluctuation et l'intervalle de confiance sont deux outils probabilistes proches dans leur forme, mais fondamentalement différents dans leur finalité. L'intervalle de fluctuation est utilisé lorsque la proportion théorique de la population est connue. Il permet de déterminer l'intervalle dans lequel la fréquence observée dans un échantillon a de fortes chances de se situer, sous l'hypothèse que le modèle théorique est correct. Il est donc essentiellement un outil de vérification d'hypothèse. L'intervalle de confiance, en revanche, est construit lorsque le paramètre de la population est inconnu. Il fournit un encadrement probabiliste autour de l'estimation, en tenant compte de l'erreur d'échantillonnage. L'intervalle de confiance vise ainsi à donner une zone plausible pour la valeur réelle d'un paramètre, avec un certain niveau de confiance (généralement 95 %).

4. Le biais dans la théorie de l'estimation correspond à la différence entre l'espérance mathématique de cet estimateur et la valeur réelle du paramètre à estimer. Lorsque cette différence est nulle, l'estimateur est dit sans biais ; dans le cas contraire, il est biaisé, ce qui signifie qu'il produit systématiquement des estimations décalées par rapport à la valeur réelle.

5. Une statistique travaillant sur la population totale est appelée statistique exhaustive. Elle correspond à un recensement, par opposition au sondage. Théoriquement, ce type de statistique ne relève plus de l'inférence puisque le paramètre est directement observable. Le développement des données massives modifie partiellement cette distinction. Dans certains domaines, on dispose aujourd'hui de volumes de données proches d'une exhaustivité. Toutefois, ces données soulèvent de nouveaux problèmes : biais de sélection, qualité variable des données, surreprésentations de certains groupes. Ainsi, même avec des masses de données, la réflexion statistique reste indispensable.

6. Le choix d'un estimateur est un enjeu central de la théorie statistique. Il détermine la qualité de l'estimation, sa précision, sa stabilité et sa fiabilité. Un bon estimateur doit présenter plusieurs propriétés : il doit être sans biais, convergent, de variance minimale et, dans certains contextes, robuste face aux valeurs aberrantes. Le compromis entre biais et variance est fondamental : un estimateur peut être légèrement biaisé mais plus précis, ou inversement. Le critère de l'erreur quadratique moyenne permet de synthétiser ces deux dimensions. Enfin, les contraintes théoriques, comme la borne de Cramér-Rao, fixent une limite inférieure à la variance des estimateurs sans biais.

7. Plusieurs grandes méthodes d'estimation existent. La plus importante est la méthode du maximum de vraisemblance, qui consiste à choisir le paramètre rendant les observations les plus probables. Elle est étroitement liée à la notion d'information de Fisher. D'autres approches reposent sur l'existence de statistiques exhaustives, qui concentrent toute l'information contenue dans l'échantillon sur le paramètre à estimer. Enfin, lorsque les données comportent des valeurs aberrantes, on privilégie des estimateurs robustes comme la médiane, les moyennes tronquées ou les M-estimateurs. Le choix d'une méthode dépend donc de la loi supposée des données, de la présence d'observations atypiques, du volume de données et du niveau de précision recherché.

8. Les tests statistiques sont au cœur de la statistique inférentielle. Ils permettent de prendre une décision probabilisée concernant une hypothèse formulée sur un paramètre de la population. Leur objectif principal est de déterminer si un écart observé est dû au hasard de l'échantillonnage ou traduit un effet réel. La construction d'un test suit une démarche rigoureuse : on formule une hypothèse nulle, on choisit une statistique de test, on fixe un seuil de risque α, puis on compare la statistique observée à une valeur critique issue d'une loi de probabilité de référence. La décision finale repose sur le rejet ou non de l'hypothèse nulle.

9. La statistique inférentielle fait l'objet de nombreuses critiques. Elle repose fortement sur des hypothèses parfois difficiles à vérifier (normalité, indépendance, homogénéité des variances). De plus, la significativité statistique peut être abusivement confondue avec une importance concrète ou pratique. Sinon, les méthodes classiques sont souvent fragiles face aux valeurs aberrantes, ce qui justifie le développement croissant d'approches robustes. Ainsi, la statistique inférentielle demeure un outil indispensable à la connaissance scientifique, mais elle doit être utilisée avec rigueur, esprit critique et conscience de ses limites.

II- Mise en œuvre Python

Question 1 “Théorie de l'échantillonnage” :

```
#Théorie de l'échantillonnage (intervalles de fluctuation)
#L'échantillonnage se base sur la répétitivité.
print("Résultat sur le calcul d'un intervalle de fluctuation")

donnees = pd.DataFrame(ouvrirUnFichier("./data/Echantillonnage-100-Echantillons.csv"))
print (donnees)

mpour = int(donnees["Pour"].mean().round())
print("moyenne Pour :", mpour)

mcontre = int(donnees["Contre"].mean().round())
print("moyenne Contre :", mcontre)

so = int(donnees["Sans opinion"].mean().round())
print("moyenne Sans opinion :", so)
```

```
PS C:\Data\Master 1 - GAED SCT\Blieck-2025-2026-Analyse-de-donnees\Seance-05\src> py main.py
Résultat sur le calcul d'un intervalle de fluctuation
    Pour  Contre  Sans opinion
0    395     396           209
1    379     432           189
2    384     426           190
3    395     407           198
4    389     413           198
..   ...     ...           ...
95   370     424           206
96   400     412           188
97   394     412           194
98   395     412           193
99   386     411           203

[100 rows x 3 columns]
moyenne Pour : 391
moyenne Contre : 416
moyenne Sans opinion : 193
```

Ces premiers résultats nous permettent de visualiser la structure moyenne des échantillons. Nous voulons maintenant comparer les moyennes avec la population mère.

```
# --- Fréquences population mère ---
Fp = round(852 / 2185, 2)
Fc = round(911 / 2185, 2)
Fso = round(422 / 2185, 2)

print("\nFréquences réelles population mère :")
print("Pour :", Fp)
print("Contre :", Fc)
print("Sans opinion :", Fso)
```

```
Fréquences réelles population mère :
Pour : 0.39
Contre : 0.42
Sans opinion : 0.19
```

```
Fréquences observées :
Pour : 0.39
Contre : 0.42
Sans opinion : 0.19
```

En comparant les fréquences de la population observée à celles de la population mère, nous pouvons constater que les résultats sont identiques, ce qui montre que la population observée est représentative de la population mère.

Nous voulons ensuite calculer des intervalles de fluctuation à 95%. C'est dans ces intervalles que devraient se situer les fréquences issues d'un échantillon.

```python
# --- Fonction intervalle de fluctuation ---
z = 1.96  # seuil de 95%
N = 2185  # taille de la population mère

def intervalle_fluctuation(f, n):
    se = math.sqrt(f * (1 - f) / n)
    bas = round(f - z * se, 4)
    haut = round(f + z * se, 4)
    return bas, haut
```

```python
# --- Intervalles pour chaque catégorie ---
IC_pour = intervalle_fluctuation(Fp, N)
IC_contre = intervalle_fluctuation(Fc, N)
IC_so = intervalle_fluctuation(Fso, N)

print("\nIntervalles de fluctuation 95% :")
print("Pour :", IC_pour)
print("Contre :", IC_contre)
print("Sans opinion :", IC_so)
```

Le terminal nous affiche alors les résultats recherchés : des intervalles qui indiquent ce que l'on peut attendre d'un échantillonnage à répétition.

```
Intervalles de fluctuation 95% :
Pour : (0.3695, 0.4105)
Contre : (0.3993, 0.4407)
Sans opinion : (0.1736, 0.2064)
```

Nous cherchons maintenant à vérifier la cohérence avec la population à travers une comparaison. Il s'agit de savoir si les valeurs réelles de la population mère se situent bien dans nos intervalles.

```
# --- Comparaison ---
print("\nComparaison :")
print("Pour : Observé =", fp_obs, " | Attendu entre", IC_pour)
print("Contre : Observé =", fc_obs, " | Attendu entre", IC_contre)
print("Sans opinion : Observé =", fso_obs, " | Attendu entre", IC_so)
```

```
Comparaison :
Pour : Observé = 0.39  | Attendu entre (0.3695, 0.4105)
Contre : Observé = 0.42  | Attendu entre (0.3993, 0.4407)
Sans opinion : Observé = 0.19  | Attendu entre (0.1736, 0.2064)
```

Les résultats obtenus suite à cette comparaison permettent ainsi d'affirmer que les échantillons sont bien représentatifs de la population mère.

Question 2 "Théorie de l'estimation" :

Pour cette deuxième partie nous nous concentrons désormais sur un cas où nous n'avons qu'un seul échantillon de la population mère à étudier. Nous commençons donc par extraire cet échantillon dont nous calculons la somme, puis les fréquences et l'intervalle de confiance de chaque opinion, tout en les comparant aux résultats précédents.

```
#Théorie de l'estimation (intervalles de confiance)
#L'estimation se base sur l'effectif.
print("Résultat sur le calcul d'un intervalle de confiance")

echantillon=donnees.iloc[0].to_list()
total=sum(echantillon)
frequence1=echantillon[0]/total
frequence2=echantillon[1]/total
frequence3=echantillon[2]/total
IC_f1=intervalle_fluctuation(frequence1,total)
IC_f2=intervalle_fluctuation(frequence2,total)
IC_f3=intervalle_fluctuation(frequence3,total)
print ("Total:",total)
print(frequence1,frequence2,frequence3)
print(IC_f1,IC_f2,IC_f3)
```

```
Résultat sur le calcul d'un intervalle de confiance
Total: 1000
0.395 0.396 0.209
(0.3647, 0.4253) (0.3657, 0.4263) (0.1838, 0.2342)
```

Avant d'interpréter les résultats obtenus, je réitère l'opération avec deux autres échantillons pour confirmer mon hypothèse. Voici les programmes et les résultats obtenus :

```
echantillon=donnees.iloc[1].to_list()
total=sum(echantillon)
frequence1=echantillon[0]/total
frequence2=echantillon[1]/total
frequence3=echantillon[2]/total
IC_f1=intervalle_fluctuation(frequence1,total)
IC_f2=intervalle_fluctuation(frequence2,total)
IC_f3=intervalle_fluctuation(frequence3,total)
print ("Total:",total)
print(frequence1,frequence2,frequence3)
print(IC_f1,IC_f2,IC_f3)

echantillon=donnees.iloc[2].to_list()
total=sum(echantillon)
frequence1=echantillon[0]/total
frequence2=echantillon[1]/total
frequence3=echantillon[2]/total
IC_f1=intervalle_fluctuation(frequence1,total)
IC_f2=intervalle_fluctuation(frequence2,total)
IC_f3=intervalle_fluctuation(frequence3,total)
print ("Total:",total)
print(frequence1,frequence2,frequence3)
print(IC_f1,IC_f2,IC_f3)
```

```
Total: 1000
0.379 0.432 0.189
(0.3489, 0.4091) (0.4013, 0.4627) (0.1647, 0.2133)
Total: 1000
0.384 0.426 0.19
(0.3539, 0.4141) (0.3954, 0.4566) (0.1657, 0.2143)
```

Ces différents résultats me permettent de constater une grande stabilité de résultats entre les échantillons étudiés et donc une bonne représentativité de la population mère malgré l'incertitude inhérente à l'échantillonnage.

Question 3 “Théorie de la décision” :

Afin de déterminer si une distribution statistique suit une loi normale, nous avons utilisé le test de Shapiro–Wilk, implémenté dans la fonction “scipy.stats.shapiro()”.

```
#Théorie de la décision (tests d'hypothèse)
#La décision se base sur la notion de risques alpha et bêta.
#Comme à la séance précédente, l'ensemble des tests se trouve au lien : https://docs.scipy.org/doc/scipy/reference/stats.html
print("Théorie de la décision")
loi1=ouvrirUnFichier("./data/Loi-normale-Test-1.csv").iloc[:,0]
loi2=ouvrirUnFichier("./data/Loi-normale-Test-2.csv").iloc[:,0]
loi1=pd.to_numeric(loi1,errors="coerce").dropna()   #tentative resoudre erreur
loi2=pd.to_numeric(loi2,errors="coerce").dropna()   #tentative resoudre erreur
stat1,res1=scipy.stats.shapiro(loi1)
stat2,res2=scipy.stats.shapiro(loi2)
print ("Loi 1:", round(stat1,4), round(res1,4))
print ("Loi 2:", round(stat2,4), round(res2,4))
```

Deux fichiers ont été analysés : Loi-normale-Test-1.csv et Loi-normale-Test-2.csv. Après application de la fonction “shapiro()” à chaque jeu de données, les résultats obtenus sont les suivants :

```
Théorie de la décision
Loi 1: 0.9639 0.0
Loi 2: 0.2609 0.0
```

Pour les deux fichiers la p.value est égale à 0 et est donc strictement inférieure au seuil nécessaire à des fichiers suivant une loi normale. Ces documents ne suivent donc pas une loi normale. Il faudra donc utiliser des tests plus adaptés.

Question bonus : Il apparaît que l’un des fichiers ne suit pas une loi normale. En s’appuyant sur les distributions analysées lors de la séance précédente, la loi la plus susceptible de modéliser ces données est la loi exponentielle. En effet, cette loi se caractérise par une distribution asymétrique et décroissante, ce qui correspond aux caractéristiques observées dans les valeurs du fichier et justifie ainsi le rejet de l’hypothèse de normalité.

Séance 6

I- Questions de cours

1. Une statistique ordinale regroupe l’ensemble des méthodes fondées sur le classement d’objets ou d’individus, c’est-à-dire sur l’ordre des observations plutôt que sur leurs valeurs numériques exactes. Elle repose sur l’ordonnancement d’une série d’observations sous forme de rangs, notés $X(1) \leq \cdots \leq X(n)$. Elle s’oppose aux statistiques nominales, qui organisent les individus en catégories sans relation d’ordre entre elles. La statistique ordinale mobilise des variables ordinales, c’est-à-dire des variables qualitatives pour lesquelles un ordre naturel peut être défini, généralement croissant, bien que certaines applications spécifiques,

comme la loi rang-taille, puissent privilégier un ordre différent. Ce type de statistique permet de représenter des hiérarchies spatiales, fréquentes en géographie. De nombreux phénomènes produisent spontanément des classements (taille des villes, intensité des crues ou des séismes, dynamisme socio-économique). L'ordination met ainsi en évidence les positions relatives des territoires, en distinguant les entités dominantes, intermédiaires ou marginales.

2. Dans les classifications, l'ordre à privilégier est l'ordre croissant, également appelé ordre naturel. Il facilite l'analyse statistique des rangs, la détection des valeurs aberrantes et l'étude de certaines propriétés des distributions, notamment l'examen des valeurs extrêmes d'une série.

3. La corrélation des rangs vise à mesurer la proximité globale entre deux séries ordonnées en comparant les rangs attribués aux mêmes individus. Elle permet d'identifier si les classements sont similaires, inverses ou indépendants, notamment à l'aide des coefficients de Spearman ou de Kendall. La concordance de classements, en revanche, repose sur l'analyse des paires de rangs. Elle évalue le nombre de paires concordantes et discordantes afin de déterminer dans quelle mesure l'ordre naturel est respecté entre deux classements. Ainsi, alors que la corrélation fournit une mesure synthétique de la relation entre deux ordres, la concordance examine la cohérence de ces ordres paire par paire.

4. Les tests de Spearman et de Kendall ont pour objectif commun de comparer des classements, mais ils reposent sur des logiques différentes. Le test de Spearman calcule une corrélation à partir des rangs, en s'appuyant sur les écarts entre les rangs de deux séries. Il est sensible à la présence d'ex æquo et, pour des échantillons de grande taille, sa distribution peut être assimilée à une loi normale. Le test de Kendall, quant à lui, se fonde sur le dénombrement des paires concordantes et discordantes. Il compare directement l'ordre relatif de chaque paire d'individus, ce qui le rend conceptuellement plus simple et plus robuste dans certains cas. Il présente également l'avantage de pouvoir être généralisé à plusieurs classements. Ainsi, de manière générale, Spearman mesure la proximité des rangs de façon quantitative, tandis que Kendall évalue la cohérence de l'ordre de manière qualitative.

5. Le coefficient de Goodman-Kruskal mesure la force de l'association d'ordre entre deux variables ordinales en comparant le nombre de paires concordantes et discordantes. Il varie entre −1 et +1 : une valeur proche de +1 indique une concordance forte, une valeur proche de −1 une inversion totale, et une valeur nulle une absence d'association apparente. Ce coefficient est conceptuellement proche du τ de Kendall. Le coefficient Q de Yule constitue un cas particulier du coefficient de Goodman-Kruskal, applicable uniquement aux tableaux de contingence 2×2. Il permet de mesurer l'association entre deux variables dichotomiques (oui/non, présent/absent). Comme Γ, il varie entre −1 et +1 et renseigne sur le sens et l'intensité de l'association. Par conséquent, le coefficient de Goodman-Kruskal fournit une mesure générale de l'association d'ordre entre variables ordinales, tandis que celui de Yule propose un outil spécifique aux situations binaires, permettant d'analyser rigoureusement les dépendances et hiérarchies observées dans les données.

II- Mise en oeuvre Python

Questions 1 à 6 : J'ai commencé par isoler la colonne « Surface (km²) » puis par la compléter par les surfaces continentales (Asie/Afrique/Europe, Amérique, Antarctique, Australie), après conversion explicite des valeurs en float, conformément aux consignes.

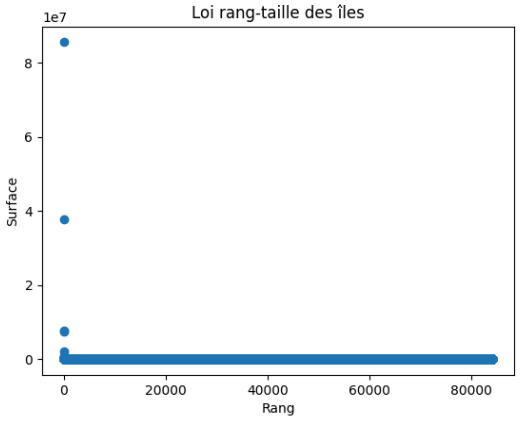
```
# Isolation de la colonne "Surface (km2)" et conversion en liste Python
surfaces = list(iles["Surface (km²)"])

# Cast en float
surfaces = [float(x) for x in surfaces if not np.isnan(x)]

# Ajout des continents
surfaces.append(float(85545323))
surfaces.append(float(37856841))
surfaces.append(float(7768030))
surfaces.append(float(7605049))
```

La liste obtenue a ensuite été ordonnée par ordre décroissant à l'aide de la fonction "ordreDecroissant()". La visualisation de la loi rang–taille met en évidence une forte dissymétrie : quelques entités concentrent l'essentiel des surfaces, tandis que la majorité possède des superficies bien plus faibles.

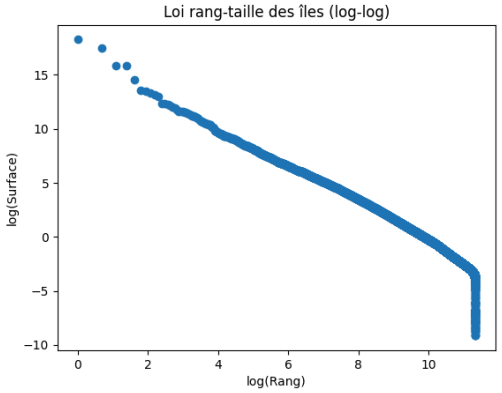
```
# Classement décroissant
surfaces = ordreDecroissant(surfaces)

# Loi rang-taille
rangs = list(range(1, len(surfaces) + 1))
```

La première représentation graphique étant illisible, les axes ont été convertis en échelle logarithmique grâce à la fonction "conversionLog()". Cette transformation permet de faire apparaître une relation quasi linéaire entre le rang et la taille, ce qui est caractéristique d'une loi rang–taille.

```
# Visualisation simple (illisible volontairement)
plt.figure()
plt.plot(rangs, surfaces, "o")
plt.title("Loi rang-taille des îles")
plt.xlabel("Rang")
plt.ylabel("Surface")
plt.savefig("./images/rang_taille_iles.png")
plt.close()
```




```python
# Conversion logarithmique
log_rangs = conversionLog(rangs)
log_surfaces = conversionLog(surfaces)

# Visualisation log-log
plt.figure()
plt.plot(log_rangs, log_surfaces, "o")
plt.title("Loi rang-taille des îles (log-log)")
plt.xlabel("log(Rang)")
plt.ylabel("log(Surface)")
plt.savefig("./images/rang_taille_iles_loglog.png")
plt.close()
```



Question 7 :

```python
# COMMENTAIRE QUESTION 7 :
# Il n'est pas possible de réaliser un test de corrélation ou de concordance des rangs
# car il n'existe ici qu'un seul classement (les surfaces). Les tests de Spearman et
# Kendall nécessitent la comparaison de deux classements distincts.
```

Questions 8 à 13 :

J'ai intégré le fichier Le-Monde-HS-Etats-du-monde-2007-2025.csv et je l'ai ouvert avec "ouvrirUnFichier()". Les colonnes suivantes ont été extraites :

- Etat
- Population 2007
- Population 2025
- Densité 2007
- Densité 2025

```
monde = ouvrirUnFichier("./data/Le-Monde-HS-Etats-du-monde-2007-2025.csv")

# Isolation des colonnes et conversion en listes Python
etat = list(monde["État"])

pop2007 = list(monde["Pop 2007"])
pop2025 = list(monde["Pop 2025"])
dens2007 = list(monde["Densité 2007"])
dens2025 = list(monde["Densité 2025"])
```

Les listes ont ensuite été ordonnées de manière décroissante à l'aide de la fonction "ordrePopulation()", en conservant l'association entre les rangs et les États. La fonction "classementPays()" m'a permis de préparer la comparaison entre les classements par population et par densité, puis les colonnes ont été isolées sous forme de listes distinctes à l'aide d'une boucle.

```
# Classements décroissants
ordre_pop2007 = ordrePopulation(pop2007, etat)
ordre_pop2025 = ordrePopulation(pop2025, etat)
ordre_dens2007 = ordrePopulation(dens2007, etat)
ordre_dens2025 = ordrePopulation(dens2025, etat)
```

```python
# Comparaison population / densité
classement_2007 = classementPays(ordre_pop2007, ordre_dens2007)
classement_2025 = classementPays(ordre_pop2025, ordre_dens2025)

# Tri selon le classement population
classement_2007.sort()
classement_2025.sort()

# Séparation en deux listes (boucle obligatoire)
rang_pop_2007 = []
rang_dens_2007 = []

for element in classement_2007:
    rang_pop_2007.append(element[0])
    rang_dens_2007.append(element[1])

rang_pop_2025 = []
rang_dens_2025 = []

for element in classement_2025:
    rang_pop_2025.append(element[0])
    rang_dens_2025.append(element[1])
```

Question 14 : Les coefficients de corrélation des rangs (Spearman) et de concordance (Kendall) ont été calculés à l'aide des fonctions "spearmanr()" et "kendalltau()" de la bibliothèque scipy.stats.

```python
# Tests statistiques
spearman_2007 = scipy.stats.spearmanr(rang_pop_2007, rang_dens_2007)
kendall_2007 = scipy.stats.kendalltau(rang_pop_2007, rang_dens_2007)

spearman_2025 = scipy.stats.spearmanr(rang_pop_2025, rang_dens_2025)
kendall_2025 = scipy.stats.kendalltau(rang_pop_2025, rang_dens_2025)

print("2007 - Spearman :", spearman_2007)
print("2007 - Kendall :", kendall_2007)
print("2025 - Spearman :", spearman_2025)
print("2025 - Kendall :", kendall_2025)
```

Voici les résultats obtenus dans mon terminal :

```
2007 - Spearman : SignificanceResult(statistic=np.float64(0.09282161580857642), pvalue=np.float64(0.2244992001066993))
2007 - Kendall : SignificanceResult(statistic=np.float64(0.0668100551149348), pvalue=np.float64(0.19192222373864458))
2025 - Spearman : SignificanceResult(statistic=np.float64(-0.026873715386233794), pvalue=np.float64(0.7092037640652133))
2025 - Kendall : SignificanceResult(statistic=np.float64(-0.007454401268834261), pvalue=np.float64(0.8770161661254758))
```

Dans les deux années étudiées, les coefficients de Spearman et de Kendall sont proches de zéro et les p-values sont largement supérieures au seuil de 5 %. On ne peut donc pas rejeter l'hypothèse nulle d'indépendance. Cela signifie que le classement des États selon leur population, et leur classement selon leur densité, sont faiblement liés, voire indépendants, aussi bien en 2007 qu'en 2025. Autrement dit, un pays très peuplé n'est pas nécessairement très dense, et inversement. Cette absence de concordance persiste dans le temps, ce qui suggère une stabilité structurelle de cette indépendance entre population totale et densité.

Question bonus : Afin de comparer le classement des îles selon leur surface et selon la longueur du trait de côte, j'ai créé une fonction locale pour calculer automatiquement les coefficients de corrélation des rangs (Spearman) et de concordance des rangs (Kendall). Cette fonction permet d'évaluer si les îles les plus étendues sont également celles dont le trait de côte est le plus long. Pour faciliter l'analyse, le code de comparaison des classements par population et par densité a été factorisé sous la forme d'une fonction générique. Cette fonction renvoie directement les coefficients nécessaires à l'interprétation statistique des classements. Un algorithme a ensuite été mis en place pour analyser la concordance des rangs pour l'ensemble des années comprises entre 2007 et 2025. Ce traitement permet de vérifier la stabilité temporelle de la relation entre les classements de population et de densité, sans répéter inutilement le code.

```
def analyseClassement(classement):
    rang1 = []
    rang2 = []
    for element in classement:
        rang1.append(element[0])
        rang2.append(element[1])
    return scipy.stats.spearmanr(rang1, rang2), scipy.stats.kendalltau(rang1, rang2)
```

```
# Exemple d'utilisation
analyse_2007 = analyseClassement(classement_2007)
analyse_2025 = analyseClassement(classement_2025)
```

Retour sur le cours

Réflexion personnelle sur les sciences des données et les humanités numériques

Ce parcours d'initiation aux sciences des données et aux humanités numériques a constitué une expérience à la fois exigeante et formatrice. Partant sans aucune connaissance préalable en informatique ou en programmation, l'apprentissage de Python s'est révélé particulièrement complexe. Les difficultés techniques rencontrées dès les premières séances, notamment lors de l'installation des outils et de la compréhension du code, ont parfois été décourageantes et ont nécessité un important investissement personnel. Néanmoins, j'ai fait le choix de persévérer tout au long du semestre, ce qui a été déterminant dans l'appropriation progressive des méthodes proposées.

Au fil des exercices, j'ai commencé à comprendre la logique du raisonnement informatique et statistique. Si la programmation m'a souvent semblé plus difficile qu'un outil comme Excel, en particulier pour une personne n'ayant jamais pratiqué ce type de démarche, elle offre néanmoins une rigueur méthodologique bien supérieure. Là où Excel permet une manipulation relativement intuitive mais parfois opaque des données, Python impose d'expliciter chaque étape du traitement, de la sélection des variables à la visualisation des résultats. Cette formalisation renforce la compréhension des méthodes statistiques mobilisées et garantit une meilleure transparence scientifique.

Ce parcours m'a également permis de mieux saisir l'intérêt des sciences des données dans le champ des humanités numériques et, plus spécifiquement, en géographie. Les outils étudiés permettent de structurer, d'analyser et de représenter des phénomènes spatiaux complexes, de mettre en évidence des tendances générales, des inégalités ou des hiérarchies territoriales, et d'interroger la relation entre données et interprétation. Les différentes mises en œuvre réalisées montrent que les données ne constituent pas une fin en soi, mais un support à la réflexion géographique, qui nécessite un regard critique et théorique.

Ainsi, cette expérience m'a permis de développer des compétences transversales essentielles, telles que la rigueur, l'autonomie et la capacité à faire face à la difficulté. Bien que les sciences des données demeurent un domaine exigeant et encore partiellement maîtrisé, cette initiation m'a permis d'en comprendre les enjeux et les apports pour la recherche en sciences humaines et sociales. Elle constitue ainsi une étape structurante de mon parcours en Master GAED-SCT, en m'ouvrant à des méthodes désormais indispensables pour analyser et comprendre les dynamiques territoriales contemporaines.