RAPPORT D'ACTIVITÉ

Analyse de données
Parcours Débutant

Manuel SEVERY
Etudiant en première année du Master GAED
Parcours SCT : Faculté des Lettres, Sorbonne Université

TABLE DES MATIÈRES:

SÉANCE 1

I- Objectifs

L'objectif de cette première séance était de mettre en place un environnement Python isolé à l'aide de Docker. L'environnement devait permettre :
- d'installer automatiquement Python (version 3.6).
- de charger les bibliothèques nécessaires à l'analyse de données (via requirements.txt).
- d'exécuter des scripts Python directement depuis le terminal à l'intérieur du conteneur Docker.

SÉANCE 2:

I. Objectifs

- Manipuler un fichier C.S.V.
- Faire des sorties graphiques
- Utiliser les bibliothèques *Pandas* (données) et *Matplotlib* (graphiques)
  N.B. *pd* et *plt* sont des alias qui remplacent respectivement *pandas* et *matplotlib.pyplot*
- Calculer des effectifs
- Calculer des fréquences
- Faire des graphiques (diagrammes en bâton et circulaires, et histogrammes)

II. Questions de cours

1) Quel est le positionnement de la géographie par rapport aux statistiques ?

L'usage de la statistique en géographie s'impose aujourd'hui comme une nécessité, bien qu'il ait longtemps suscité des réticences. Héritière d'une approche interprétative issue des

sciences humaines, la géographie a souvent relégué la rigueur mathématique au second plan. Cette distance s'explique à la fois par une tradition qualitative et par la formation inégale des géographes face aux méthodes quantitatives, parfois source d'erreurs d'interprétation.

Pourtant, la masse d'informations générée par la discipline — qu'il s'agisse de données spatiales, socio-économiques ou environnementales — exige désormais des outils capables d'en révéler la structure et la cohérence. Les méthodes statistiques répondent à ce besoin : elles offrent au géographe les moyens d'organiser, de représenter et de modéliser la complexité du réel.

En intégrant ces outils, la géographie transforme sa démarche. De science descriptive, elle tend vers une analyse explicative fondée sur la recherche de régularités et de relations à différentes échelles. L'essor de l'analyse spatiale illustre cette évolution vers une pratique plus scientifique, où la mesure et la modélisation deviennent partie intégrante de la compréhension des dynamiques territoriales.

2) Le hasard existe-t-il en géographie ?

Aborder le hasard en géographie suppose d'articuler réflexion philosophique et pratique scientifique. D'un point de vue conceptuel, deux visions s'opposent : celle du déterminisme, pour laquelle tout événement découle d'une cause identifiable, et celle d'un probabilisme qui reconnaît l'existence du hasard tout en l'attribuant à des causes encore non élucidées.

Sur le plan empirique, la géographie statistique admet l'aléa comme une composante incontournable des phénomènes spatiaux. Si chaque situation locale échappe à la prévision absolue, l'observation d'un grand nombre de cas permet de dégager des régularités à l'échelle des ensembles. Cette perspective repose sur la distinction entre un « hasard bénin », régi par des distributions comme la loi normale, et un « hasard sauvage », marqué par des événements extrêmes que modélisent des lois à queue lourde, de type paretien.

Ainsi, l'analyse géographique ne nie pas le hasard : elle le mesure, le traduit en probabilités et en tendances globales, transformant l'incertitude individuelle en connaissance statistique des dynamiques spatiales.

3) Quels sont les types d'information géographique ?

On retrouve deux types d'information géographique:
D'une part, les données attributaires fournissent des indications sur les caractéristiques des entités spatiales : elles concernent la population, les revenus, les paramètres environnementaux ou tout autre indicateur socio-économique.
D'autre part, les données géométriques décrivent la configuration spatiale des objets étudiés, qu'il s'agisse de points, de lignes ou de polygones.

4) Quels sont les besoins de la géographie au niveau de l'analyse de données ?

La géographie doit :

- Collecter et structurer les données à partir de nomenclatures cohérentes et de métadonnées complètes (définitions, sources, dates, modalités d'observation) pour assurer la fiabilité et la comparabilité des informations.

- Analyser les structures internes à l'aide d'outils statistiques variés : corrélations, régressions, analyses factorielles, classifications, identification des lois de probabilité et construction de modèles explicatifs ou prédictifs.

- Visualiser les résultats sous forme de cartes, diagrammes, graphiques ou modèles spatiaux afin de représenter les dynamiques territoriales.

- Interpréter les résultats en les confrontant aux connaissances disciplinaires et aux conditions de collecte pour éviter les conclusions biaisées.

L'objectif est de transformer la description des phénomènes en une compréhension des logiques spatiales qui les organisent.

5) Quelles sont les différences entre la statistique descriptive et la statistique explicative ?

- Statistique descriptive **:** elle constitue la première étape de l'analyse. Son objectif est de résumer, organiser et représenter les données observées afin d'en faire ressortir les principales tendances. Elle mobilise des indicateurs de position et de dispersion (moyenne, médiane, écart-type), identifie les valeurs atypiques et s'appuie sur des représentations visuelles telles que diagrammes, histogrammes, boîtes à moustaches ou cartes thématiques. En géographie, elle permet de décrire la répartition de la population, les densités ou les caractéristiques d'un territoire sans chercher à en expliquer les causes.

- Statistique explicative (ou inférentielle) **:** elle intervient dans un second temps pour modéliser et comprendre les relations entre variables. Elle vise à établir des liens de causalité ou de dépendance — par exemple entre développement économique et accessibilité, ou entre urbanisation et dynamiques migratoires. Ses principaux outils sont les analyses de corrélation, les régressions simples ou multiples, les analyses factorielles, les tests statistiques, ou encore les modèles généralisés. En géographie, elle sert à expliquer les processus spatiaux ou à prévoir des évolutions à partir de modèles probabilistes.

En somme, la statistique descriptive observe et synthétise tandis que la statistique explicative relie et interprète : l'une décrit les faits, l'autre cherche à comprendre leurs mécanismes.

6) Quelles sont les types de visualisation de données en géographie ? Comment choisir celles-ci ?

La visualisation des données géographiques repose sur un ensemble de représentations adaptées à la nature des variables et aux objectifs d'analyse :

- Variables qualitatives (catégorielles) : diagrammes en barres, diagrammes en secteurs (ou camemberts), représentations horizontales ou comparatives adaptées à la mise en évidence des parts relatives.

- Variables quantitatives discrètes : diagrammes en bâtons permettant d'observer la fréquence de chaque modalité.

- Variables quantitatives continues : histogrammes, polygones de fréquence, courbes cumulatives ou boîtes à moustaches pour analyser la distribution, la dispersion et les valeurs extrêmes.

- Données spatiales : cartes thématiques ou choroplèthes pour représenter la répartition géographique, nuages de points et cartes de chaleur pour visualiser les corrélations et les concentrations.

- Analyses multivariées : graphes factoriels issus d'ACP ou d'AFC, utiles pour représenter les proximités et les structures de données dans l'espace des variables.

Le choix d'une visualisation dépend donc à la fois du type de variable (qualitative, quantitative, discrète ou continue) et du but poursuivi : comparer des groupes, décrire une distribution, détecter des anomalies, représenter des corrélations ou révéler des structures spatiales complexes.

7) Quelles sont les méthodes d'analyse de données possibles ?

Les méthodes utilisées en géographie statistique peuvent être regroupées en trois grandes catégories complémentaires :

- Méthodes descriptives et multidimensionnelles : elles servent à résumer, explorer et structurer l'information. L'analyse en composantes principales (ACP) permet de réduire la dimensionnalité et de représenter les relations entre variables quantitatives. L'analyse factorielle des correspondances (AFC) et l'analyse des

correspondances multiples (ACM) s'appliquent aux variables qualitatives, tandis que les approches mixtes (AFDM, AFM) traitent des jeux de données hétérogènes. Les classifications — hiérarchiques ou par nuées dynamiques — facilitent le regroupement d'individus ou de variables selon leurs similarités, que l'on peut également cartographier par des analyses de proximités.

- Méthodes explicatives : elles visent à établir des relations causales entre variables. On y trouve la régression simple ou multiple pour les variables quantitatives, l'ANOVA et les modèles linéaires généraux, ainsi que la régression logistique ou l'analyse discriminante pour les variables qualitatives. Les techniques de segmentation permettent d'élaborer des typologies ou de distinguer des profils caractéristiques.

- Méthodes de prévision : elles s'appuient sur les modèles de séries temporelles (autorégressifs, entre autres) afin d'estimer l'évolution future d'un phénomène à partir de ses valeurs passées.

8) Comment définiriez-vous : (a) population statistique ? (b) individu statistique ? (c) caractères statistiques ? (d) modalités statistiques ? Quels sont les types de caractères ? Existe-t-il une hiérarchie entre eux ?

(a) Population statistique : elle correspond à l'ensemble des unités sur lesquelles porte l'étude. Il peut s'agir, par exemple, de l'ensemble des habitants d'une région, des entreprises d'un secteur ou des communes d'un territoire.

(b) Individu statistique : c'est l'unité élémentaire composant la population. En géographie, on parle souvent d'unité spatiale lorsqu'elle peut être localisée (une ville, une commune, un quartier, un salarié, etc.).

(c) Caractères statistiques : ils désignent les propriétés mesurées ou observées pour chaque individu, telles que l'âge, le revenu, la superficie, l'altitude ou la catégorie socio-professionnelle.

(d) Modalités statistiques : ce sont les différentes valeurs ou catégories que peut prendre un caractère. Les modalités doivent être à la fois exclusives (un individu ne peut appartenir qu'à une seule modalité) et exhaustives (toutes les possibilités sont prises en compte).

Types de caractères :

- Qualitatifs : ils expriment une catégorie. Ils peuvent être *nominales* (sans ordre, comme le type d'habitat ou la profession) ou *ordinales* (avec un ordre, comme les niveaux de diplôme).

- Quantitatifs : ils traduisent une mesure numérique. Ils peuvent être *discrets* (valeurs entières isolées, comme le nombre d'enfants) ou *continus* (valeurs comprises dans un intervalle, comme le revenu ou l'âge). On distingue également les *variables d'intervalle* et les *variables de rapport* selon la valeur et l'interprétation du zéro.

Il n'existe pas de hiérarchie de valeur entre ces types de caractères. En revanche, du point de vue spatial, on distingue les unités primaires, correspondant aux données non agrégées (les « atomes » de l'analyse), et les unités secondaires, issues d'agrégations (les « molécules »), distinction essentielle pour l'étude des échelles et la modélisation spatiale.

9) Comment mesurer une amplitude et une densité ?

Amplitude:

- L'amplitude d'une classe d'intervalle]a,b]]a,b] correspond à sa longueur.
- Elle se calcule par la différence entre la borne supérieure et la borne inférieure : A=b−a
- Elle sert à caractériser la largeur de chaque classe dans une série continue.

Densité:

- La densité d'une classe met en rapport l'importance de la classe et sa largeur.
- Pour une classe d'effectif nini et d'amplitude b−ab−a, la densité est donnée par d=n(i)/b−a
- Elle permet de comparer des classes d'amplitudes différentes en tenant compte à la fois de leur effectif et de leur étendue, ce qui est essentiel pour interpréter correctement un histogramme de classes inégales.

10) A quoi servent les formules de Sturges et de Yule ?

Ces formules servent à guider le choix du nombre de classes k lors de la discrétisation d'une variable continue (préparation d'un histogramme), et visent à éviter un découpage trop fin (bruit) ou trop grossier (perte d'information):

- Sturges : $k \approx 1 + 3{,}2222 \times \log 10(n)$
- Yule : $k \approx 2.5(n)1/4$

11) Comment définir un effectif ? Comment calculer une fréquence et une fréquence cumulée ? Qu'est-ce qu'une distribution statistique ?

- Effectif (n_i) : nombre d'occurrences d'une modalité.
- Fréquence (f_i) : part relative de cette modalité dans la population totale. Calcul : f(*i*) = n(*i*)/n

- Fréquence cumulée : somme progressive des fréquences jusqu'à une modalité donnée. Calcul :

$$F_k = \sum_{i=1}^{k} f_i$$

- Distribution statistique : répartition des effectifs (ou fréquences) selon les modalités d'un caractère. La distribution statistique fait le lien observable entre les lois de probabilité et les données, servant donc de base à la description.

III. Mise en oeuvre avec Python

Étapes 1 à 5:

Une fois les fichiers de la Séance-02 téléchargés (dossier src > dossier data > fichier resultats-elections-presidentielles-2022-1er-tour.csv / fichier main.py) , on se situe dans le dossier de la séance avec la commande (cmd):

```
C:\Users\anne_\Desktop\Forriez-2025-2026-Analyse-de-donnees\Forriez-2025-2026-Analyse-de-donnees> cd Seance-02
```

Dans le fichier main.py se trouve le code:

```python
#coding:utf8

import pandas as pd
import matplotlib.pyplot as plt

# Source des données : https://www.data.gouv.fr/datasets/election-presidentielle-des-10-et-24-avril-2022-resultats-definitifs-du-1er-tou
with open("./data/resultats-elections-presidentielles-2022-1er-tour.csv","r") as fichier:
    contenu = pd.read_csv(fichier)

# Mettre dans un commentaire le numéro de la question
# Question 1
# ...
```

On peut alors afficher sur le terminal exécutant le conteneur la variable *contenu* avec le code suivant:

```python
# Mettre dans un commentaire le numéro de la question
# Question 1
# ...
print(contenu)
```

Ce qui permet d'afficher sur le terminal:

```
    Code du département          Libellé du département  Inscrits  ...         Nom.11  Prénom.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            Polynésie française    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-Barthélemy     24414  ...  DUPONT-AIGNAN    Nicolas    339.0
106                  ZZ  Français établis hors de France   1435746  ...  DUPONT-AIGNAN    Nicolas   7074.0

[107 rows x 56 columns]

C:\Users\anne_\Desktop\Forriez-2025-2026-Analyse-de-donnees\Forriez-2025-2026-Analyse-de-donnees\Seance-02>
```

Etapes 6 à 7:

On calcule le nombre de lignes et de colonnes du tableau de données avec le code len suivant:

```python
# Question 6
print("Nombre de lignes :", len(contenu))
print("Nombre de colonnes:", len(contenu.columns))
```

Ce qui affiche sur le terminal:

```
[107 rows x 56 columns]
Nombre de lignes : 107
Nombre de colonnes: 56

C:\Users\anne_\Desktop\Forriez-2025-2026-Analyse-de-donnees\Forriez-2025-2026-Analyse-de-donnees\Seance-02>
```

On utilise le code suivant pour afficher sur le terminal les types détectés par Pandas:

```python
#Question 7
print(contenu.dtypes)
```

Ce qui nous donne, de gauche à droite, la liste du type de chaque colonne (en utilisant les fonctions *int*, *float*, *str* et *bool*) :

```
Nombre de colonnes: 56
Code du département        object
Libellé du département     object
Inscrits                    int64
Abstentions               float64
Votants                   float64
Blancs                    float64
Nuls                      float64
Exprimés                  float64
Sexe                       object
Nom                        object
Prénom                     object
Voix                      float64
Sexe.1                     object
Nom.1                      object
Prénom.1                   object
Voix.1                    float64
Sexe.2                     object
Nom.2                      object
Prénom.2                   object
Voix.2                    float64
Sexe.3                     object
Nom.3                      object
Prénom.3                   object
Voix.3                    float64
Sexe.4                     object
Nom.4                      object
Prénom.4                   object
Voix.4                    float64
Sexe.5                     object
Voix.4                    float64
Sexe.5                     object
Nom.5                      object
Prénom.5                   object
Voix.5                    float64
Sexe.6                     object
Nom.6                      object
Prénom.6                   object
Voix.6                    float64
Sexe.7                     object
Nom.7                      object
Prénom.7                   object
Voix.7                    float64
Sexe.8                     object
Nom.8                      object
Prénom.8                   object
Voix.8                    float64
Sexe.9                     object
Nom.9                      object
Prénom.9                   object
Voix.9                    float64
Sexe.10                    object
Nom.10                     object
Prénom.10                  object
Voix.10                   float64
Sexe.11                    object
Nom.11                     object
Prénom.11                  object
Voix.11                   float64
dtype: object
```

Nous pouvons ainsi faire le point sur la nature statistique des variables:

1) *Object (str* en langage python): il désigne des chaînes de caractères, et il s'agit de variables qualitatives nominales (texte).
2) *int64 (int)*: il désigne des entiers, et il s'agit de variables quantitatives discrètes (nombre).
3) *float64 (float)*: il désigne les nombres réels, et il s'agit de variables quantitatives continues (proportions, pourcentages etc)
4) *bool* : il désigne une valeur de validité (True / False).

Etapes 8 à 10:

Pour afficher sur le terminal le nombre de colonnes avec la méthode Pandas head, on peut utiliser le code suivant:

```
#Question 8
print("Nom des colonnes:")
print(contenu.head)
```

Ce qui nous donne comme résultat sur le terminal:

```
Nom des colonnes:
<bound method NDFrame.head of     Code du département          Libellé du département  Inscrits  Abstentions  ...  Sexe.11        N
om.11  Prénom.11  Voix.11
0                   01                           Ain    438109      97541.0  ...        M  DUPONT-AIGNAN    Nicolas   8998.0
1                   02                         Aisne    373544     101089.0  ...        M  DUPONT-AIGNAN    Nicolas   5790.0
2                   03                        Allier    249991      58497.0  ...        M  DUPONT-AIGNAN    Nicolas   4216.0
3                   04          Alpes-de-Haute-Provence    128075      29290.0  ...        M  DUPONT-AIGNAN    Nicolas   2504.0
4                   05                  Hautes-Alpes    113519      25357.0  ...        M  DUPONT-AIGNAN    Nicolas   2142.0
..                 ...                           ...       ...          ...  ...      ...            ...        ...      ...
102                 ZP              Polynésie française    205576     142121.0  ...        M  DUPONT-AIGNAN    Nicolas   1969.0
103                 ZS          Saint-Pierre-et-Miquelon      5045       2272.0  ...        M  DUPONT-AIGNAN    Nicolas     82.0
104                 ZW                 Wallis et Futuna      9528       4125.0  ...        M  DUPONT-AIGNAN    Nicolas    244.0
105                 ZX     Saint-Martin/Saint-Barthélemy     24414      15812.0  ...        M  DUPONT-AIGNAN    Nicolas    339.0
106                 ZZ  Français établis hors de France   1435746     931455.0  ...        M  DUPONT-AIGNAN    Nicolas   7074.0
```

A l'aide du nom des colonnes affichées, on peut sélectionner le nombre des inscrits avec le code:

```
#Question 9
print("Colonne des inscrits:")
print(contenu.Inscrits)
```

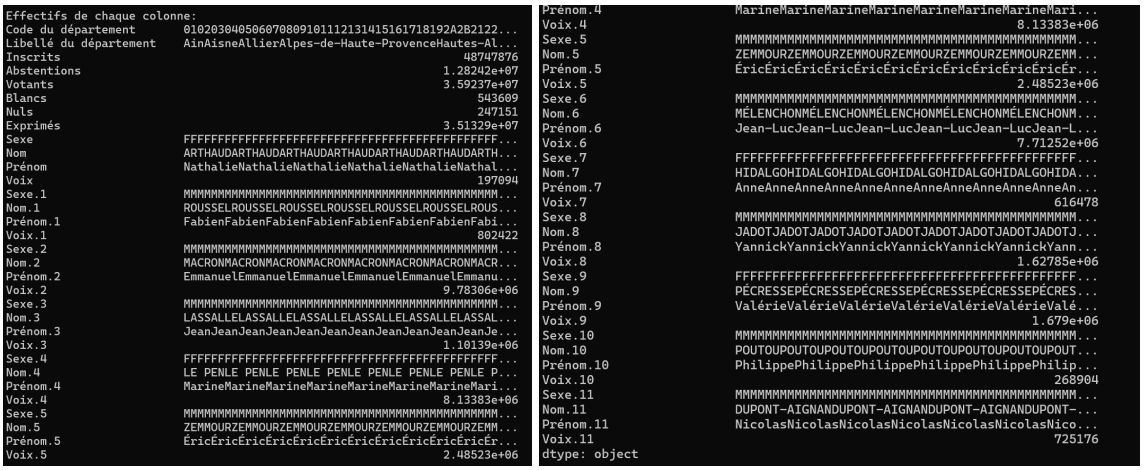
Ce qui nous donne comme résultat sur le terminal:

```
Colonne des inscrits:
0       438109
1       373544
2       249991
3       128075
4       113519
        ...
102     205576
103       5045
104       9528
105      24414
106    1435746
Name: Inscrits, Length: 107, dtype: int64
```

Avec la méthode Pandas sum (...), on peut calculer les effectifs de chaque colonne et les placer dans une liste (à l'aide d'une boucle) avec le code:

```python
#Question 10
print("Effectifs de chaque colonne:")
print(contenu.sum())
```

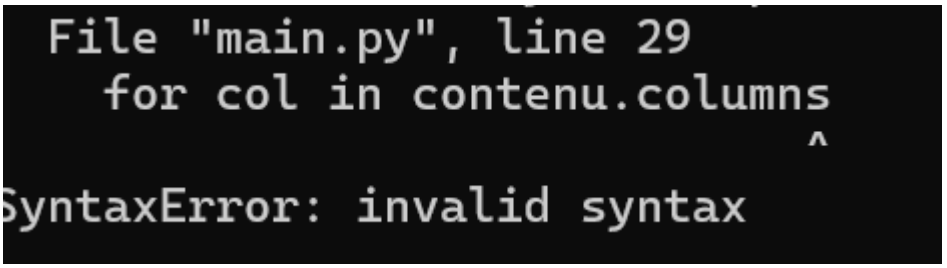
Ce qui donne pour étrange résultat (de gauche à droite):



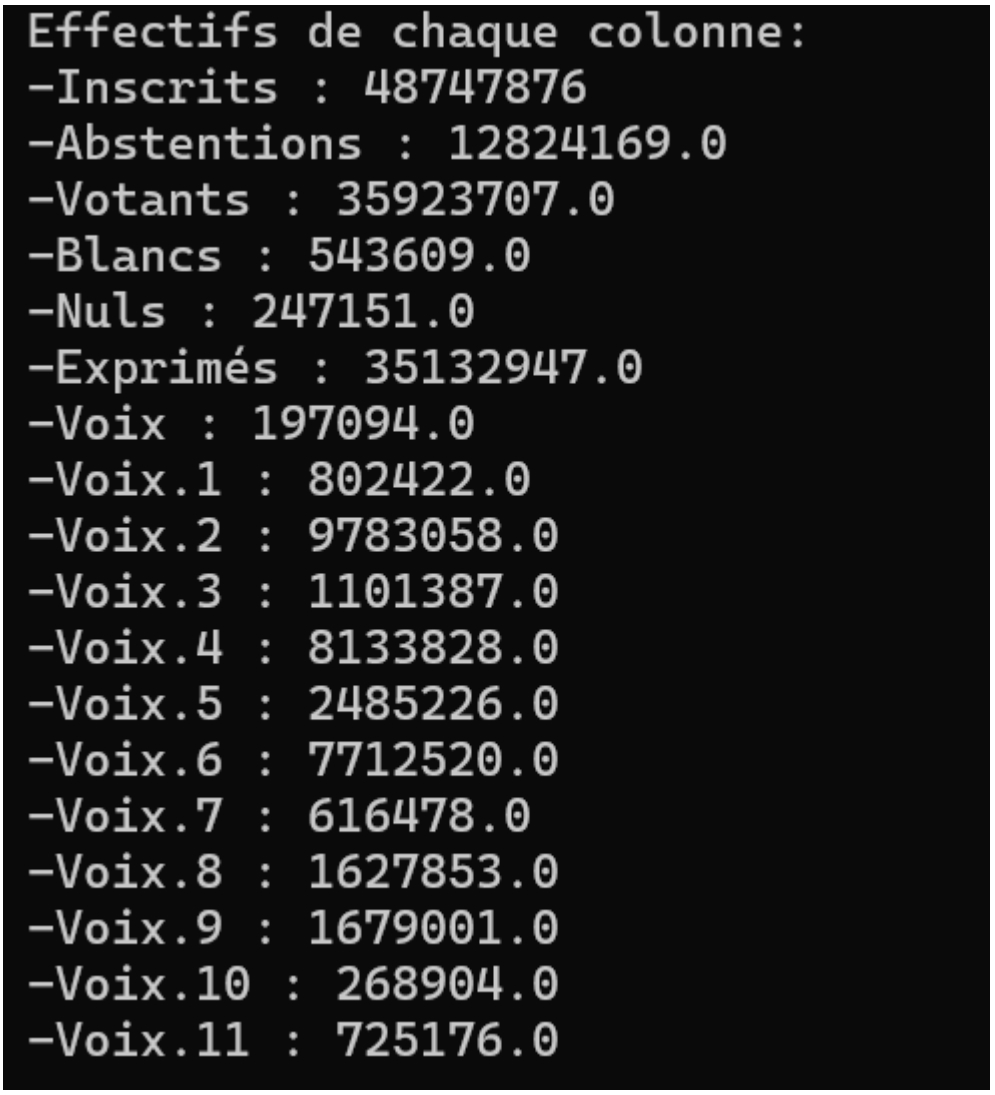
On ajoute alors la condition suivante pour ne sélectionner que les valeurs quantitatives (ce qui enlèvera les valeurs textuelles), en utilisant le code:

```python
#Question 10
#Création d'une liste vide
somme_colonnes = []
#Condition
for col in contenu.columns
    if contenu[col].dtypes in ["int64" , "float64"]:
        total = contenu[col].sum()
        somme_colonnes.append((col, total))
#Résultat
print("Effectifs de chaque colonne:")
for col, total in somme_colonnes:
    print(f"-{col} : {total}")
```

Lors de ma première tentative, le résultat a échoué à cause de la syntaxe:



En modifiant la ligne avec l'ajout du : , on peut afficher sur le terminal le résultat suivant:

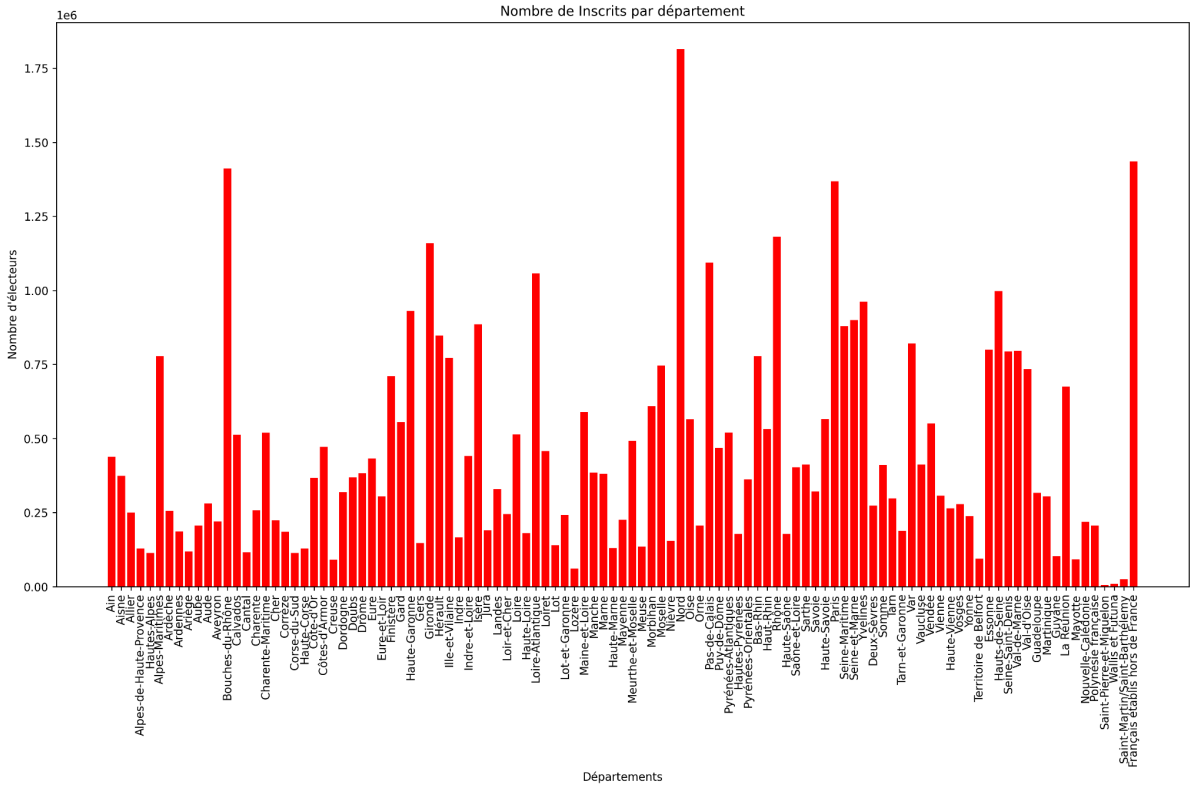


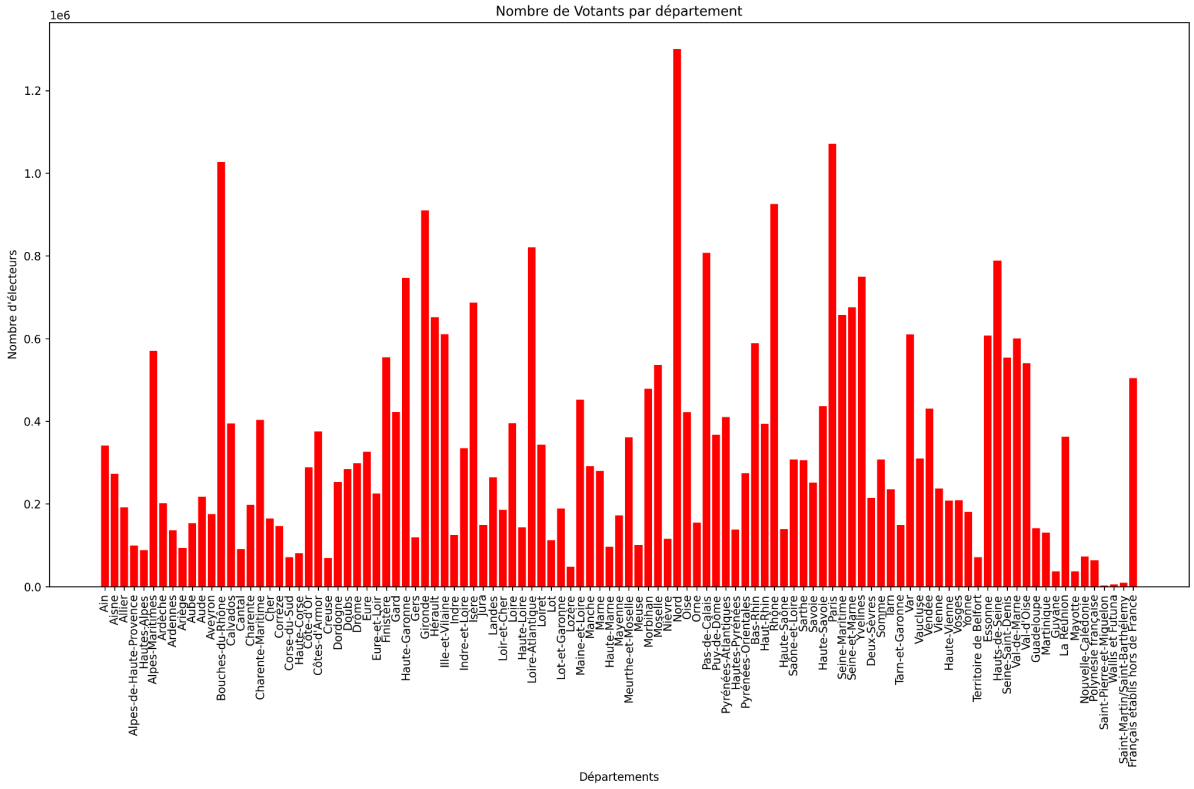
Etapes 11 à 13:

Pour faire des diagrammes en barres avec le nombre des inscrits et le nombre des votants pour chaque département, il faut en premier lieu créer un dossier images avec os:

```python
#Question 11
#Création du dossier
import os
os.makedirs("images", exist_ok=True)
colonnes = ["Inscrits" , "Votants"]
for col in colonnes:
    plt.figure(figsize=(15,10))
    plt.bar(contenu["Libellé du département"], contenu[col], color="red")
    plt.xticks(rotation=90)
    plt.title(f"Nombre de {col} par département")
    plt.ylabel("Nombre d'électeurs")
    plt.xlabel("Départements")
    plt.tight_layout()
    plt.savefig(f"images/{col}.png", dpi=250)
    plt.close()
```

Le programme donné fait apparaître les graphiques suivant :

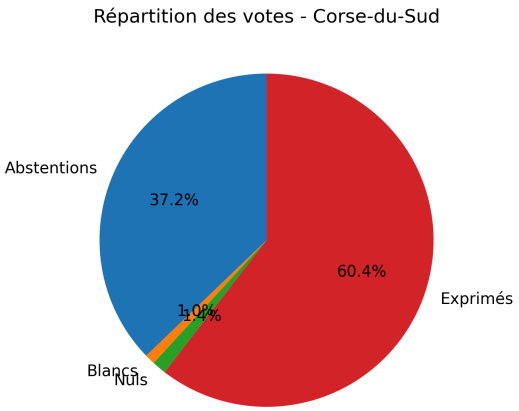


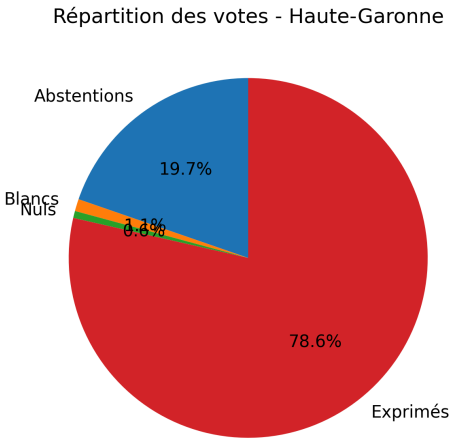


Pour l'étape 12, je commence par créer un dossier images pour réceptionner les graphiques:

```python
#Question 12
#Création du dossier
import os
os.makedirs("./images_pie", exist_ok=True)
colonnes = ["Abstentions", "Blancs", "Nuls", "Exprimés"]
for idx, row in contenu.iterrows():
    valeurs = [row[col] for col in colonnes]
    labels = colonnes
    plt.pie(valeurs, labels=labels, autopct='%1.1f%%', startangle=90)
    plt.title(f"Répartition des votes - {row['Libellé du département']}")
    plt.savefig(f"images_pie/{row['Code du département']}_{row['Libellé du département']}.png", dpi=300)
    plt.close()
print("Diagrammes circulaires")
```

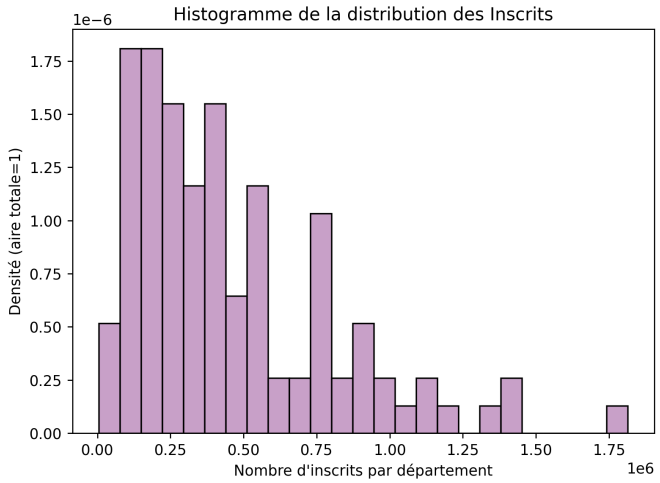
Ce qui donne pour résultat dans le dossier images:





Pour l'étape 13, je dois créer un histogramme de la colonne "Inscrits", en commençant par créer un dossier pour réceptionner les images:

```
#Question 13
import os
os.makedirs("images_histogramme", exist_ok=True)
plt.hist(contenu["Inscrits"], bins=25, color="#C8A2C8", edgecolor="black", density=True)
plt.title("Histogramme de la distribution des Inscrits")
plt.xlabel("Nombre d'inscrits par département")
plt.ylabel("Densité (aire totale=1)")
plt.tight_layout()
plt.savefig("images_histogramme/histogramme_inscrits.png", dpi=250)
plt.close()
print("Histogramme")
```



SÉANCE 3

I- Partie Cours

1) Quel caractère est le plus général : le caractère quantitatif ou le caractère qualitatif ? Justifier pourquoi.

Le caractère qualitatif est le plus général. Un caractère qualitatif permet de décrire une population par des catégories ou modalités, sans qu'il soit nécessaire d'associer des nombres, par exemple la couleur des yeux, la profession, le type de sol ou le genre. Un caractère quantitatif, lui, est un cas particulier de caractère qualitatif : il correspond à des modalités qui peuvent être mesurées et exprimées par des nombres, comme l'âge, la taille, le revenu ou la distance.Tout caractère quantitatif peut être vu comme qualitatif si l'on regroupe ses valeurs numériques en classes (par exemple, âge : 0–10 ans, 10–20 ans, etc.), mais l'inverse n'est pas toujours possible, car beaucoup de caractères qualitatifs ne peuvent pas être mesurés numériquement.

2) Quels sont les caractères quantitatifs discrets et caractères quantitatifs continus ? Pourquoi les distinguer ?

- Les caractères quantitatifs discrets sont ceux dont les valeurs sont dénombrables, c'est-à-dire qui ne peuvent prendre qu'un nombre fini ou dénombrable de valeurs possibles : nombre d'enfants, nombre de logements, nombre de communes, etc.
- Les caractères quantitatifs continus, eux, peuvent prendre toute valeur dans un intervalle réel, comme la température, la superficie, le revenu ou la distance.

On les distingue parce qu'ils n'impliquent pas les mêmes méthodes de traitement statistique : un caractère discret se traite par des comptages et fréquences, tandis qu'un caractère continu nécessite des classes (ou intervalles) pour regrouper les valeurs, car elles sont infinies. Cette distinction est donc essentielle pour choisir la bonne représentation graphique (histogramme ou diagramme en bâtons) et les bons paramètres de synthèse.

3) Paramètres de position. Pourquoi existe-t-il plusieurs types de moyenne ? Pourquoi calculer une médiane ? Quand est-il possible de calculer un mode?

Il existe plusieurs types de moyenne (arithmétique, géométrique, harmonique, pondérée) parce qu'elles ne traduisent pas la même réalité statistique. La moyenne arithmétique s'applique à des valeurs homogènes et additives (par exemple, un revenu moyen), la moyenne géométrique est utilisée pour des évolutions relatives (croissance, indices), et la moyenne harmonique s'emploie pour des vitesses ou des ratios.

La médiane représente la valeur centrale d'une série ordonnée : elle sépare la population en deux effectifs égaux. On la calcule lorsque la distribution est asymétrique ou contient des valeurs extrêmes, car elle résume mieux la tendance centrale que la moyenne dans ce cas.

Le mode correspond à la valeur la plus fréquente. On peut le calculer seulement lorsque la variable présente une ou plusieurs modalités qui se répètent, et il s'applique aussi bien à des caractères qualitatifs (la catégorie la plus fréquente) qu'à des caractères quantitatifs (la valeur la plus représentée).

4) Paramètres de concentration. Quel est l'intérêt de la médiale et de l'indice de C. Gini ?

La médiale est une valeur qui partage la surface totale d'un histogramme en deux aires égales ; elle permet de mesurer la concentration d'une distribution, notamment lorsqu'on étudie la répartition d'une ressource (ex. : revenu, richesse).

L'indice de Gini est un indicateur synthétique de l'inégalité de répartition : il mesure la distance entre la distribution réelle et une distribution parfaitement égalitaire, variant entre 0 (égalité parfaite) et 1 (inégalité totale). Plus il est proche de 1, plus la concentration est forte (par exemple, une minorité détient une grande part de la ressource).

Ces deux indicateurs servent donc à évaluer la justice spatiale ou sociale d'un phénomène.

5) Paramètres de dispersion. Pourquoi calculer une variance à la place de l'écart à la moyenne ? Pourquoi la remplacer par l'écart type ? Pourquoi calculer l'étendue ? À quoi sert-il de

créer un quantile ? Quel(s) est (sont) le(s) quantile(s) le(s) plus utilisé(s) ? Pourquoi construire une boîte de dispersion ? Comment l'interpréter ?

Les paramètres de dispersion décrivent l'hétérogénéité des valeurs autour de la moyenne d'une distribution. L'écart simple à la moyenne additionne des écarts positifs et négatifs qui peuvent se compenser, ce qui conduit à un bilan nul malgré une forte variabilité. Pour éviter cette compensation, on calcule la variance, en élevant ces écarts au carré, ce qui fournit une mesure globale de dispersion mais exprimée dans une unité au carré, souvent peu intuitive. On lui préfère donc l'"écart" type, obtenu en prenant la racine carrée de la variance, qui ramène la mesure dans l'unité d'origine de la variable et la rend plus facilement interprétable.

L'étendue, définie comme la différence entre la valeur maximale et la valeur minimale, donne une indication rapide de l'amplitude totale des valeurs. Elle est très simple à obtenir mais demeure fortement influencée par les valeurs extrêmes, ce qui explique qu'on l'utilise surtout en complément d'indicateurs plus robustes comme l'écart interquartile.

Les quantiles découpent la distribution en parts égales afin d'analyser comment se répartissent les valeurs et d'identifier dispersion, concentration et positions relatives. On emploie particulièrement la médiane (quantile 0,5), les quartiles (Q1, Q2, Q3) ainsi que, selon le niveau de détail souhaité, les déciles ou les centiles, par exemple pour repérer les 10% d'unités les plus élevées.

La boîte de dispersion, ou boîte à moustaches (boxplot), offre une représentation graphique des quartiles, de la médiane et des valeurs extrêmes. La boîte s'étend de Q1 à Q3 avec la médiane tracée à l'intérieur, tandis que les moustaches signalent les valeurs les plus basses et les plus hautes. Ce dispositif visuel permet de juger d'un coup d'œil de la dispersion, de l'étendue et de la symétrie de la série, de détecter des valeurs aberrantes et de comparer plusieurs groupes de données ; une boîte resserrée traduit une faible dispersion, alors qu'une boîte décalée ou très allongée suggère une asymétrie ou la présence de valeurs atypiques.

6) Paramètres de forme. Quelle différence faites-vous entre les moments centrés et les moments absolus ? Pourquoi les utiliser ? Pourquoi vérifier la symétrie d'une distribution et comment faire ?

Les paramètres de forme décrivent la configuration générale d'une distribution statistique, en particulier sa symétrie et la manière dont les valeurs se concentrent autour de la moyenne. Les moments centrés sont calculés à partir des écarts à la moyenne en conservant leur signe, alors que les moments absolus prennent la valeur absolue de ces écarts, ce qui supprime toute compensation entre valeurs positives et négatives. Les moments d'ordre supérieur fournissent des informations spécifiques : le moment d'ordre 2 correspond à la variance (dispersion), le moment d'ordre 3 à l'asymétrie (skewness) et le moment d'ordre 4

à l'aplatissement ou concentration (kurtosis) de la courbe. Les moments absolus, en ignorant le signe, servent également à analyser la forme de la distribution et à repérer des asymétries ou concentrations particulières.

Vérifier la symétrie d'une distribution est essentiel, car cela conditionne le choix de l'indicateur de tendance centrale. Lorsque la distribution est symétrique, moyenne, médiane et mode sont proches ; en cas d'asymétrie à droite, on observe généralement moyenne > médiane > mode, tandis qu'en cas d'asymétrie à gauche, moyenne < médiane < mode. Une forte asymétrie tire la moyenne vers les valeurs extrêmes, ce qui peut rendre la médiane plus pertinente pour représenter la tendance centrale. La symétrie peut être examinée visuellement à l'aide d'un histogramme ou d'une boîte de dispersion (boîte à moustaches), et quantifiée numériquement par un coefficient d'asymétrie, par exemple celui de Pearson.

II- Partie python

- *Etape 1 à 4*

Dans le dossier src, je crée un dossier data et j'y introduis le fichier resultats-elections-presidentielles-2022-1er-tour.csv disponible dans la Séance-03 du GitHub.J'y introduis aussi le fichier main.py.

- *Étape 5 à 6*

Je place d'abord une condition pour sélectionner les caractères quantitatifs (#condition), avant de calculer sous forme de listes les moyennes de chaque colonne, les médianes de chaque colonne; les modes de chaque colonne ; l'écart type de chaque colonne ; l'écart absolu à la moyenne de chaque colonne ; l'étendue de chaque colonne.

```
#Etape 5
#Condition
import os
contenu = pd.read_csv(
    "data/resultats-elections-presidentielles-2022-1er-tour.csv",
    sep=None,
    engine="python",
    encoding="utf-8"
)

colonnes_quanti = ["Inscrits", "Votants", "Blancs", "Nuls", "Exprimés", "Abstentions"]
num = contenu[colonnes_quanti].copy()
```

```
#Calcul
moyennes = num.mean(axis=0).round(2)
medianes = num.median(axis=0).round(2)
modes = num.mode().iloc[0].round(2)
ecarts_type = num.std(ddof=0, axis=0).round(2)          # écart-type population[web:12]
ecarts_abs_moy = num.apply(lambda s: np.abs(s - s.mean()).mean()).round(2)
etendues = (num.max() - num.min()).round(2)
#Affichage
print("\nMoyennes :\n", moyennes)
print("\nMédianes :\n", medianes)
print("\nModes :\n", modes)
print("\nÉcarts type :\n", ecarts_type)
print("\nÉcarts absolus moyens :\n", ecarts_abs_moy)
print("\nÉtendues :\n", etendues)
```

Pour l'étape 6, j'affiche la liste des paramètres sur le terminal avec le programme suivant:

```
# Etape 6 - Construire un tableau récapitulatif et l'afficher en colonnes
resume = pd.DataFrame({
    "Moyenne": moyennes,
    "Médiane": medianes,
    "Mode": modes,
    "Écart-type": ecarts_type,
    "Écart abs. moyen": ecarts_abs_moy,
    "Étendue": etendues
})

print("\n--- Paramètres (par colonne quantitative) ---\n")
print(resume.to_string())
```

Ce qui donne sur le terminal:

```
Moyennes :
 Inscrits       455587.63
Votants        335735.58
Blancs           5080.46
Nuls             2309.82
Exprimés       328345.30
Abstentions    119852.05
dtype: float64

Médianes :
 Inscrits       366859.0
Votants        274372.0
Blancs           4001.0
Nuls             2039.0
Exprimés       268568.0
Abstentions     95369.0
dtype: float64

Modes :
 Inscrits       5045.0
Votants        2773.0
Blancs         4577.0
Nuls             17.0
Exprimés       2701.0
Abstentions    2272.0
Name: 0, dtype: float64

Écarts type :
 Inscrits       349359.73
Votants        257183.52
Blancs           3476.17
Nuls             1494.35
```

```
 Inscrits       349359.73
Votants        257183.52
Blancs           3476.17
Nuls             1494.35
Exprimés       252570.01
Abstentions    116469.70
dtype: float64

Écarts absolus moyens :
 Inscrits       272240.72
Votants        201517.17
Blancs           2817.95
Nuls             1131.99
Exprimés       197762.20
Abstentions     74959.07
dtype: float64

Étendues :
 Inscrits       1808861.0
Votants        1297100.0
Blancs           17389.0
Nuls              8236.0
Exprimés       1272080.0
Abstentions     929183.0
dtype: float64
```

```
--- Paramètres (par colonne quantitative) ---

               Moyenne   Médiane    Mode  Écart-type  Écart abs. moyen    Étendue
Inscrits     455587.63  366859.0  5045.0   349359.73         272240.72  1808861.0
Votants      335735.58  274372.0  2773.0   257183.52         201517.17  1297100.0
Blancs         5080.46    4001.0  4577.0     3476.17           2817.95    17389.0
Nuls           2309.82    2039.0    17.0     1494.35           1131.99     8236.0
Exprimés     328345.30  268568.0  2701.0   252570.01         197762.20  1272080.0
Abstentions  119852.05   95369.0  2272.0   116469.70          74959.07   929183.0
```

- *Etape 7 à 8:*

Je calcule la distance interquartile et interdécile de chaque colonne quantitative avec le programme suivant, avant d'utiliser la variable print pour afficher les résultats.

```
#Etape 7
#Calcul des distances interquartile et interdécile
Q1 = num.quantile(0.25)
Q3 = num.quantile(0.75)
dist_interquartile = (Q3 - Q1).round(2)

D1 = num.quantile(0.10)
D9 = num.quantile(0.90)
dist_interdecile = (D9 - D1).round(2)
#Affichage sous forme de colonnes
print("\n--- Distances interquartiles (par colonne) ---\n")
print(dist_interquartile.to_string())

print("\n--- Distances interdéciles (par colonne) ---\n")
print(dist_interdecile.to_string())
```

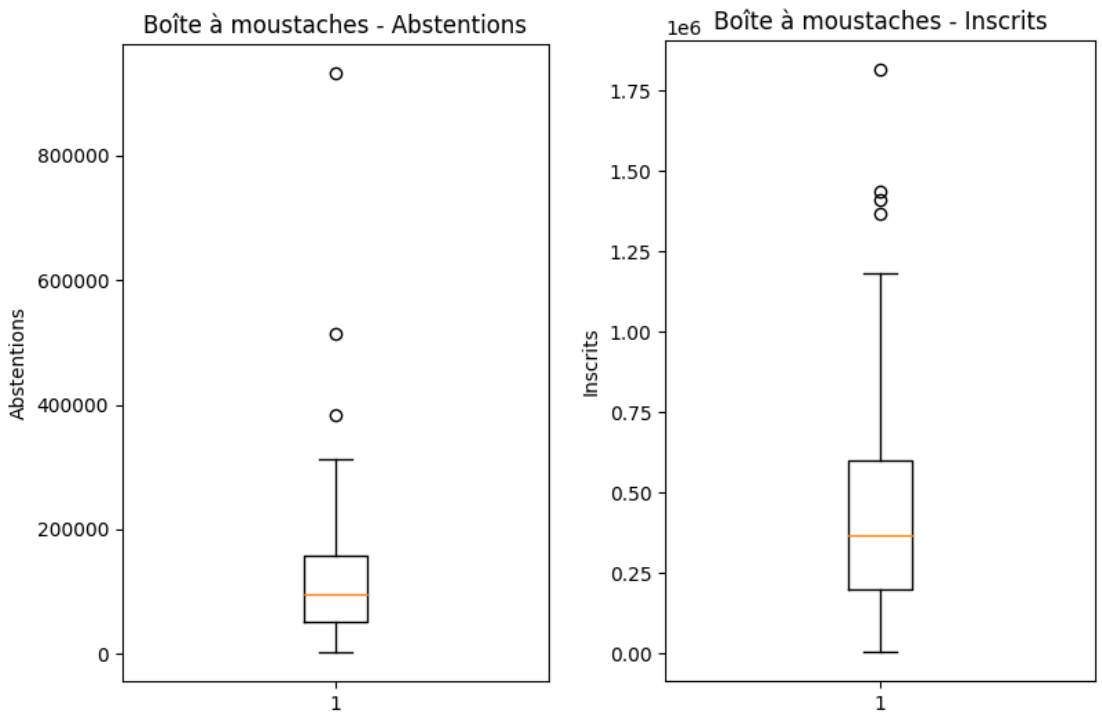
Ce qui donne sur le terminal:

```
--- Distances interquartiles (par colonne) ---

Inscrits       401050.0
Votants        301770.5
Blancs           4852.5
Nuls             1917.0
Exprimés       296870.5
Abstentions    106489.0

--- Distances interdéciles (par colonne) ---

Inscrits       793988.8
Votants        602687.2
Blancs           8845.8
Nuls             3240.6
Exprimés       590169.2
Abstentions    193676.2
```

Pour l'étape 8, à l'aide de Matplolib et d'une boucle, il faut faire des boîtes à moustache de chaque colonne quantitative. Pour ce faire, je crée d'abord le dossier image intitulé "img", avant d'écrire le programme suivant pour utiliser une boucle et enfin enregistrer l'image dans le dossier créé.

```python
#Etape 8
#Créer le dossier img
os.makedirs("img", exist_ok=True)

# Boucle sur chaque colonne quantitative
for col in num.columns:
    plt.figure(figsize=(4, 6))
    plt.boxplot(num[col].dropna(), vert=True)
    plt.title(f"Boîte à moustaches - {col}")
    plt.ylabel(col)

    # Enregistrer l'image dans le dossier img
    plt.savefig(f"img/boxplot_{col}.png", bbox_inches="tight")
    plt.close()
```

Ce qui donne dans le dossier:



Etapes 9 à 10:
Dans le dossier src, il faut d'abord introduire le dossier data le fichier island-index.csv disponible dans la Seance-03 du GitHub. Je dois écrire un algorithme pour catégoriser et
dénombrer le nombre d'îles ayant une surface comprise :
entre 0 et 10 km2 ou]0, 10] ;
entre 10 et 25 km2 ou]10, 25] ;
entre 25 et 50 km2 ou]25, 50] ;

entre 50 et 100 km2 ou]50, 100] ;
entre 100 et 2 500 km2 ou]100, 2500] ;
entre 2 500 et 5 000 km2 ou]2500, 5000] ;
entre 5 000 et 10 000 km2 ou]5000, 10000] ;
supérieur ou égal 10 000 km2 ou]10000, +∞[.

Je dois donc d'abord lire le fichier avec la méthode read_csv, avant de sélectionner la surface, définir les bornes de classes et les étiquettes. Je catégorise ensuite les surfaces:

```
    plt.close()
#Etape 10
#Lecture du fichier
df_island = pd.read_csv("data/island-index.csv", encoding="utf-8")
#Sélection de la colonne Surface (km2)
surfaces = df_island["Surface (km²)"]
#Définition des bornes de classes
bins = [0, 10, 25, 50, 100, 2500, 5000, 10000, float("inf")]
#Étiquettes des classes
labels = [
    "]0, 10]",
    "]10, 25]",
    "]25, 50]",
    "]50, 100]",
    "]100, 2500]",
    "]2500, 5000]",
    "]5000, 10000]",
    "]10000, +∞["
]
#Catégorisation des surfaces
classes = pd.cut(surfaces, bins=bins, labels=labels, right=True, include_lowest=False)
#Dénombrement des îles par classe
compte_classes = classes.value_counts().sort_index()

print("\nNombre d'îles par intervalle de surface :\n")
print(compte_classes.to_string())
```

Ce qui affiche sur le terminal:

```
Nombre d'îles par intervalle de surface :

]0, 10]          78423
]10, 25]          2327
]25, 50]          1164
]50, 100]          788
]100, 2500]       1346
]2500, 5000]        60
]5000, 10000]       40
]10000, +∞[         71
```

SÉANCE 4:

I-Questions de cours

1) **Quels critères mettriez-vous en avant pour choisir entre une distribution statistique avec des variables discrètes et une distribution statistique avec des variables continues ?**

Pour choisir entre une distribution statistique à variables discrètes et une à variables continues, plusieurs critères se distinguent clairement .

‘a)Nature de la variable
Les variables discrètes se bornent à des valeurs isolées, dénombrables ou finies, telles que le nombre d’habitants, d’accidents ou de naissances, et s’adaptent aux lois de Bernoulli, binomiale ou Poisson avec une répartition en paliers brusques .
Les variables continues couvrent un intervalle infini de valeurs, comme l’altitude, la température, le revenu ou une durée, modélisées par des densités probabilistes (normale, exponentielle, uniforme, log-normale, Weibull) .

(b)Forme empirique des données
Une allure en marches d’escalier ou en points distincts signale le discret, tandis qu’une courbe fluide et lisse évoque le continu .

(c)Processus génératif du phénomène
Les cumuls d’effets indépendants tendent vers la normale, les multiplications vers la log-normale, les comptages rares vers Poisson ; l’échelle ou la précision des mesures peut simuler une discrétisation par arrondi .

(d)Autres aspects de la distribution
La symétrie, la dispersion autour de la moyenne et le nombre de paramètres requis guident l’ajustement au réel observé, entre comptage ponctuel (discret) et mesure ininterrompue (continu) .

2) **Expliquez selon vous quelles sont les lois les plus utilisées en géographie ?**

En géographie, diverses lois statistiques s'avèrent essentielles pour modéliser la répartition, la hiérarchie et la dynamique des phénomènes spatiaux.

(a)Loi de Zipf (ou rang-taille)

Elle illustre la hiérarchie urbaine en reliant inversement la taille d'une ville à son rang, révélant l'inégalité structurelle des systèmes de peuplement.

(b)Loi log-normale (ou de Gibrat)
Elle décrit les croissances proportionnelles, comme l'évolution des populations ou des revenus, typique des phénomènes multiplicatifs spatiaux.

(c)Loi de Pareto
Elle capture les distributions à queue lourde, soulignant fortes inégalités ou concentrations où peu d'entités dominent, dans les tailles de villes ou richesses territoriales.

(d)Loi de Poisson
Elle quantifie les occurrences rares et indépendantes sur un territoire, telles que séismes, inondations, crimes, accidents ou points d'intérêt, pour analyser fréquences aléatoires.

(e)Loi normale (ou de Gauss)
Elle prédomine pour variables continues issues de multiples facteurs indépendants, comme température, pluviométrie, revenu moyen, densité ou erreurs de mesure, formant une courbe en cloche centrée.

(f)Autres lois complémentaires
La loi exponentielle modélise les temps d'attente en mobilité ou trafic, reliant ainsi formes observées à mécanismes sous-jacents de structure, fréquence et hiérarchie spatiale.

II- Mise en pratique

Tout d'abord, je télécharge le fichier main.py à partir du dossier src de la séance 4. Je crée aussi en amont un dossier pour réceptionner les représentations graphiques générées, en important d'abord:

```
#Première étape
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import os
```

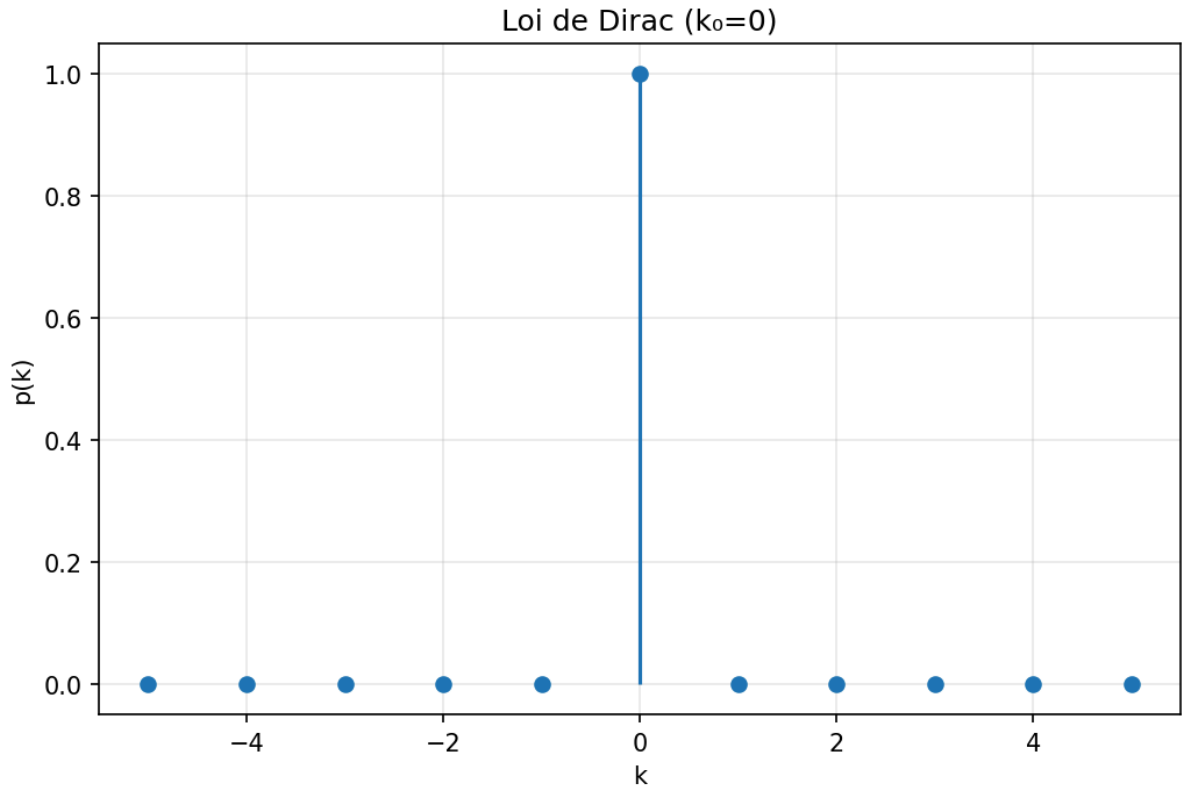
Puis j'utilise le code suivant pour créer un dossier images dans mon dossier src, afin de réceptionner les graphiques crées:

```
# Dossier img
IMG_DIR = "img"
os.makedirs(IMG_DIR, exist_ok=True)
#Code
```

Maintenant, je peux représenter les distributions statistiques de variables discrètes suivantes : Dirac, uniforme discrète, binomiale, Poisson et Zipf-Mandelbrot. Je commence par la loi Dirac avec le programme de codes suivant:

```
#Code
def plot_dirac(k0=0, kmin=None, kmax=None, save="img/dirac.png"):
    if kmin is None: kmin = k0 - 5
    if kmax is None: kmax = k0 + 5
    k = np.arange(kmin, kmax + 1)
    pmf = (k == k0).astype(int)

    plt.figure(figsize=(8,5))
    plt.stem(k, pmf, use_line_collection=True, basefmt=" ")
    plt.title(f"Loi de Dirac (k₀={k0})")
    plt.xlabel("k"); plt.ylabel("p(k)")
    plt.grid(True, alpha=0.3)
    plt.savefig(save, dpi=150, bbox_inches='tight')
    plt.show()
    plt.close()

# Utilisation
plot_dirac(k0=0, save="img/dirac.png")
```

Le programme utilisé représente de manière mathématique plus exactement la loi Dirac, montrant parfaitement la masse ponctuelle et étant paramétrable (kmin etc). Au départ, j'ai utilisé un histogramme l'approximation gaussienne, ce qui faussait la représentation. J'ai ensuite corrigé mon programme pour une meilleure représentation graphique de la loi Dirac.
Ce qui affiche comme graphique dans le dossier" img":



Je peux donc utiliser les programmes suivants pour afficher les distributions statistiques de variables discrètes suivantes : la loi uniforme discrète ; la loi binomiale ; la loi de Poisson ; la loi de Zipf-Mandelbrot. Ce qui donne les programmes suivants:

```python
#LOI UNIFORME DISCRÈTE (Diagramme en bâtons (bar))
def plot_uniforme_discrete(n=10, save="uniforme_discrete.png"):
    k = np.arange(n)
    pmf = np.ones(n) / n  # Probabilité égale 1/n

    plt.figure(figsize=(8,5))
    plt.bar(k, pmf, alpha=0.7, color='skyblue', edgecolor='navy', width=0.8)
    plt.title(f'Loi uniforme discrète (0 à {n-1})')
    plt.xlabel('k'); plt.ylabel('P(X=k)')
    plt.grid(True, alpha=0.3)
    plt.savefig(f'{IMG_DIR}/{save}', dpi=150, bbox_inches='tight')
    plt.show()
    plt.close()
```

```python
#LOI BINOMIALE (stem plot (tiges) + histogramme échantillons)
def plot_binomiale(n=20, p=0.5, size=1000, save="binomiale.png"):
    k = np.arange(n+1)
    pmf = stats.binom.pmf(k, n, p)
    samples = stats.binom.rvs(n, p, size=size)

    plt.figure(figsize=(10,6))
    plt.stem(k, pmf, basefmt=" ", linefmt='r-', markerfmt='ro', label='PMF théorique')
    plt.hist(samples, bins=range(n+2), density=True, alpha=0.6, color='orange', label='Échantillons')
    plt.title(f'Loi binomiale B({n},{p})')
    plt.xlabel('k'); plt.ylabel('P(X=k)')
    plt.legend(); plt.grid(True, alpha=0.3)
    plt.savefig(f'{IMG_DIR}/{save}', dpi=150, bbox_inches='tight')
    plt.show()
    plt.close()
```

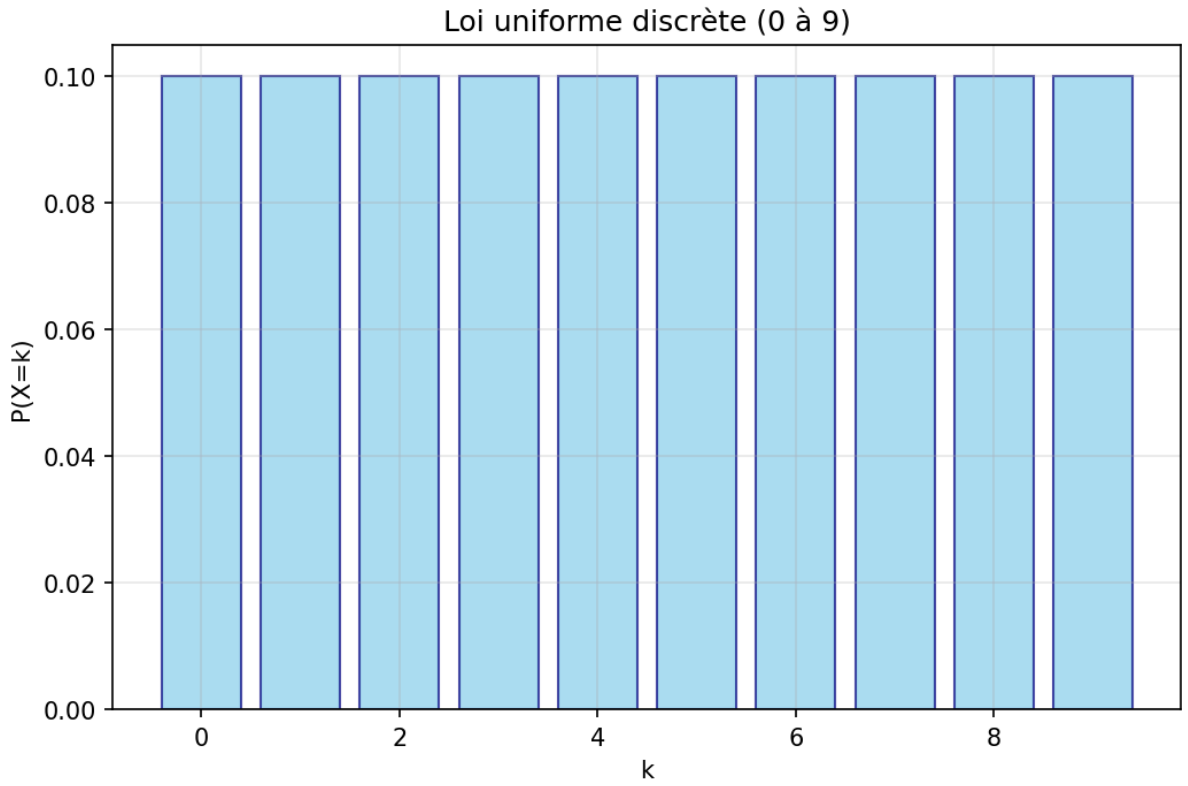
```python
#LOI DE POISSON (Stem plot (tiges) + histogramme)
def plot_poisson(mu=5, size=1000, save="poisson.png"):
    k_max = int(mu + 4*np.sqrt(mu))
    k = np.arange(k_max+1)
    pmf = stats.poisson.pmf(k, mu)
    samples = stats.poisson.rvs(mu, size=size)

    plt.figure(figsize=(10,6))
    plt.stem(k, pmf, basefmt=" ", linefmt='g-', markerfmt='go', label='PMF théorique')
    plt.hist(samples, bins=range(k_max+2), density=True, alpha=0.6, color='lightgreen', label='Échantillons')
    plt.title(f'Loi de Poisson λ={mu}')
    plt.xlabel('k'); plt.ylabel('P(X=k)')
    plt.legend(); plt.grid(True, alpha=0.3)
    plt.savefig(f'{IMG_DIR}/{save}', dpi=150, bbox_inches='tight')
    plt.show()
    plt.close()
```

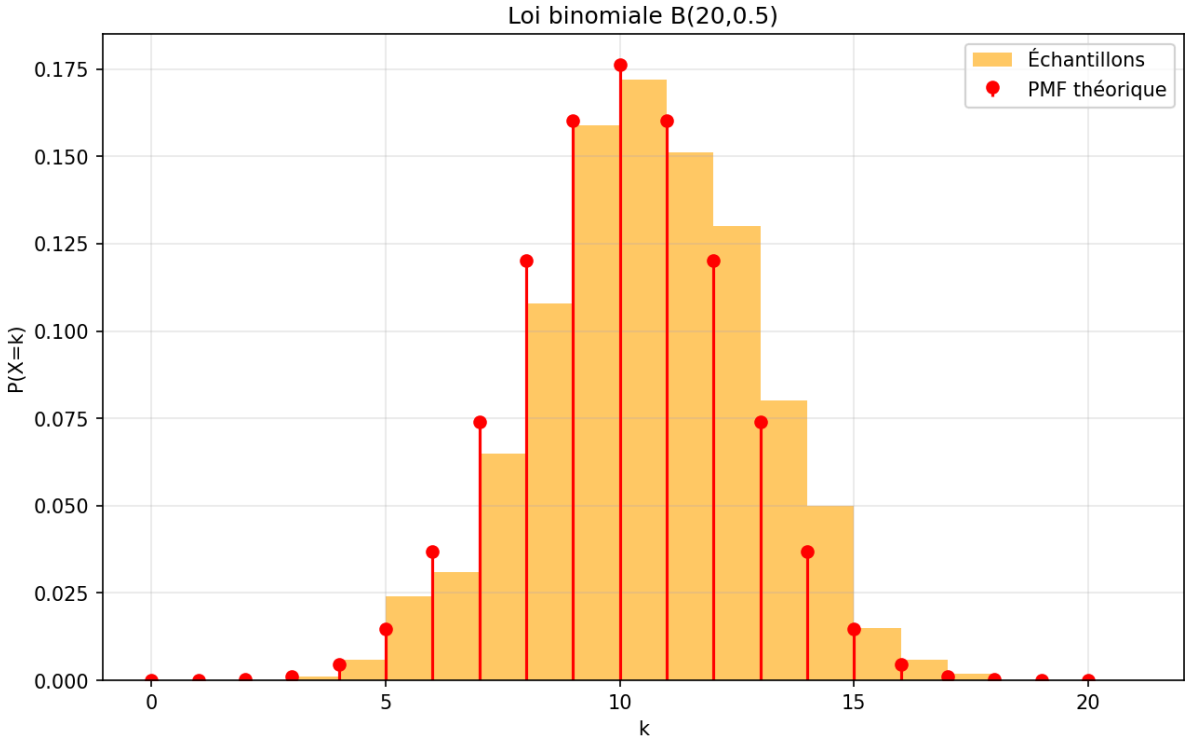
```python
#LOI ZIPF-MANDELBROT (Diagramme en bâtons décroissant (queue lourde))
def plot_zipf_mandelbrot(n_max=50, alpha=1.5, s=0.5, save="zipf_mandelbrot.png"):
    k = np.arange(1, n_max+1)
    pmf = k ** (-alpha + s)
    pmf = pmf / pmf.sum()  # Normalisation

    plt.figure(figsize=(10,6))
    plt.bar(k, pmf, alpha=0.7, color='purple', edgecolor='darkviolet', width=0.8)
    plt.title(f'Loi Zipf-Mandelbrot (α={alpha}, s={s})')
    plt.xlabel('k (rang)'); plt.ylabel('P(X=k)')
    plt.yscale('log')  # Échelle log pour voir la queue lourde
    plt.grid(True, alpha=0.3)
    plt.savefig(f'{IMG_DIR}/{save}', dpi=150, bbox_inches='tight')
    plt.show()
    plt.close()
```

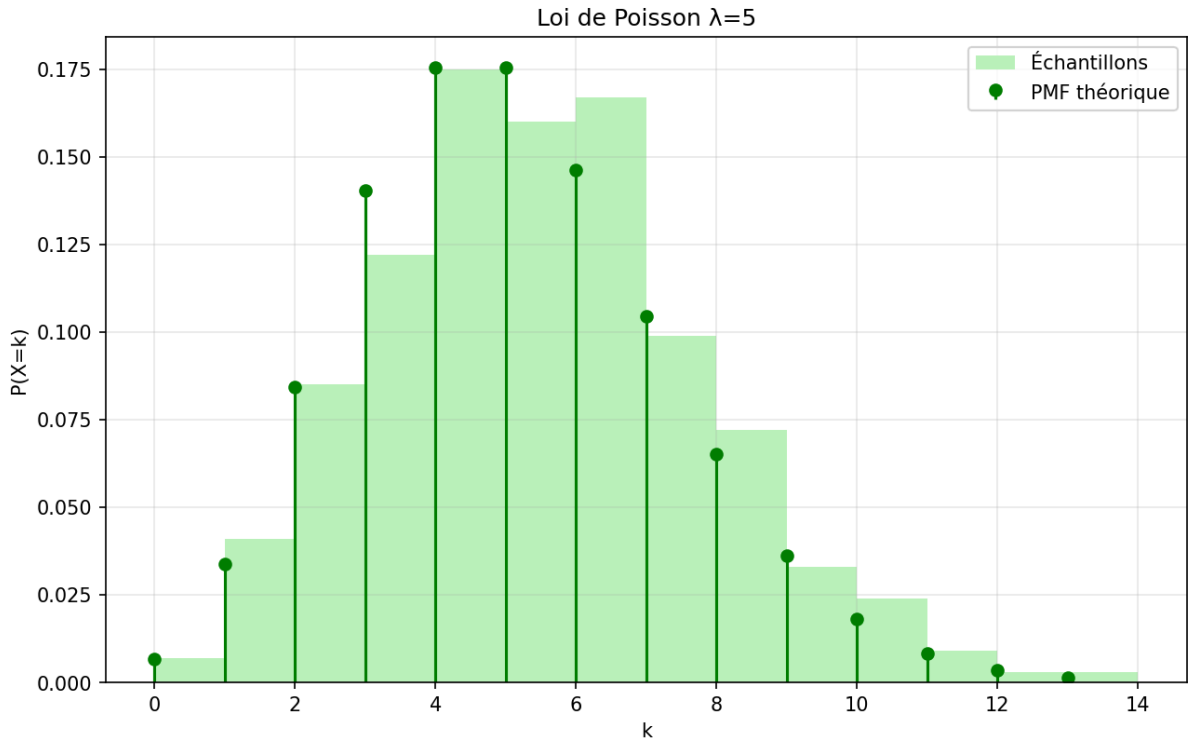
La première fois que j'ai activé ces programmes sur mon terminal, rien n'a été crée dans mon dossier. J'ai du rajouter à la fin le programme suivant pour générer les représentations graphiques, car les nouvelles fonctions n'avaient pas été appelé:

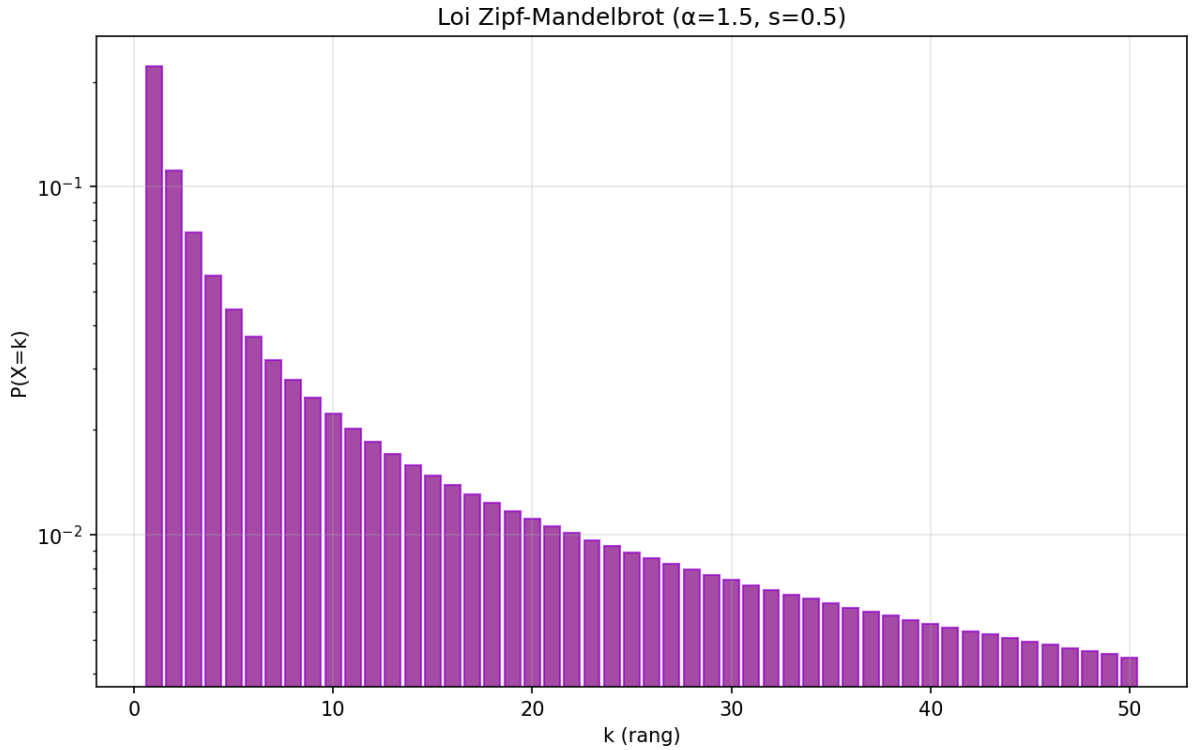
```python
#Affichage
plot_uniforme_discrete(n=10, save="uniforme_discrete.png")
plot_binomiale(n=20, p=0.5, save="binomiale.png")
plot_poisson(mu=5, save="poisson.png")
plot_zipf_mandelbrot(n_max=50, save="zipf_mandelbrot.png")
```

Ce qui donne les représentations suivantes:


Loi uniforme discrète (0 à 9)
P(X=k)
0.10
0.08
0.06
0.04
0.02
0.00
0
2
4
6
8
k


Loi binomiale B(20,0.5)
Échantillons
PMF théorique
P(X=k)
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
5
10
15
20
k





Pour les distributions statistiques de variables continues suivantes : la loi de Poisson ; la loi normale ; la loi log-normale ; la loi uniforme ; la loi du χ2 ; la loi de Pareto. J'utilise les programmes suivants :

```python
# 1. LOI DE POISSON CONTINUE ((approximation Gamma) (Histogramme + PDF))
def plot_poisson_continue(mu=5, size=1000, save="poisson_continue.png"):
    samples = stats.gamma.rvs(a=mu+1/3, scale=1, size=size)
    x = np.linspace(0, 20, 1000)
    pdf = stats.gamma.pdf(x, a=mu+1/3, scale=1)

    plt.figure(figsize=(10,6))
    plt.hist(samples, bins=50, density=True, alpha=0.6, color='lightcoral', label='Échantillons')
    plt.plot(x, pdf, 'r-', lw=3, label=f'Gamma(μ+1/3={mu+1/3})')
    plt.title(f'Loi de Poisson continue (approximation Γ)')
    plt.xlabel('x'); plt.ylabel('Densité f(x)')
    plt.legend(); plt.grid(True, alpha=0.3)
    plt.savefig(f'{IMG_DIR}/{save}', dpi=150, bbox_inches='tight')
    plt.show()
    plt.close()
```

```python
# 2. LOI NORMALE (Histogramme + PDF + intervalles confiance)
def plot_normale(mu=0, sigma=1, size=1000, save="normale.png"):
    samples = np.random.normal(mu, sigma, size)
    x = np.linspace(mu-4*sigma, mu+4*sigma, 1000)
    pdf = stats.norm.pdf(x, mu, sigma)

    plt.figure(figsize=(10,6))
    plt.hist(samples, bins=50, density=True, alpha=0.6, color='skyblue', label='Échantillons')
    plt.plot(x, pdf, 'b-', lw=3, label=f'N({mu},{sigma})')
    plt.axvline(mu, color='red', linestyle='--', label='μ')
    plt.axvline(mu+sigma, color='orange', linestyle='--', alpha=0.7, label='μ+σ')
    plt.title('Loi normale (Gaussienne)')
    plt.xlabel('x'); plt.ylabel('Densité f(x)')
    plt.legend(); plt.grid(True, alpha=0.3)
    plt.savefig(f'{IMG_DIR}/{save}', dpi=150, bbox_inches='tight')
    plt.show()
    plt.close()
```

```python
# 3. LOI LOG-NORMALE (Histogramme + PDF (échelle log optionnelle))
def plot_lognormale(mu=0, sigma=0.5, size=1000, save="lognormale.png"):
    samples = stats.lognorm.rvs(s=sigma, scale=np.exp(mu), size=size)
    x = np.linspace(0.1, 5, 1000)
    pdf = stats.lognorm.pdf(x, s=sigma, scale=np.exp(mu))

    plt.figure(figsize=(10,6))
    plt.hist(samples, bins=50, density=True, alpha=0.6, color='gold', label='Échantillons')
    plt.plot(x, pdf, 'orange', lw=3, label=f'LogN(μ={mu},σ={sigma})')
    plt.title('Loi log-normale')
    plt.xlabel('x'); plt.ylabel('Densité f(x)')
    plt.legend(); plt.grid(True, alpha=0.3)
    plt.savefig(f'{IMG_DIR}/{save}', dpi=150, bbox_inches='tight')
    plt.show()
    plt.close()
```

```python
# 4. LOI UNIFORME (Histogramme plat + PDF rectangulaire)
def plot_uniforme(a=0, b=10, size=1000, save="uniforme_continue.png"):
    samples = np.random.uniform(a, b, size)
    x = np.linspace(a-1, b+1, 1000)
    pdf = stats.uniform.pdf(x, a, b-a)

    plt.figure(figsize=(10,6))
    plt.hist(samples, bins=50, density=True, alpha=0.6, color='lightgreen', label='Échantillons')
    plt.plot(x, pdf, 'g-', lw=3, label=f'U[{a},{b}]')
    plt.fill_between(x, pdf, alpha=0.2, color='green')
    plt.title('Loi uniforme continue')
    plt.xlabel('x'); plt.ylabel('Densité f(x)')
    plt.legend(); plt.grid(True, alpha=0.3)
    plt.savefig(f'{IMG_DIR}/{save}', dpi=150, bbox_inches='tight')
    plt.show()
    plt.close()
```

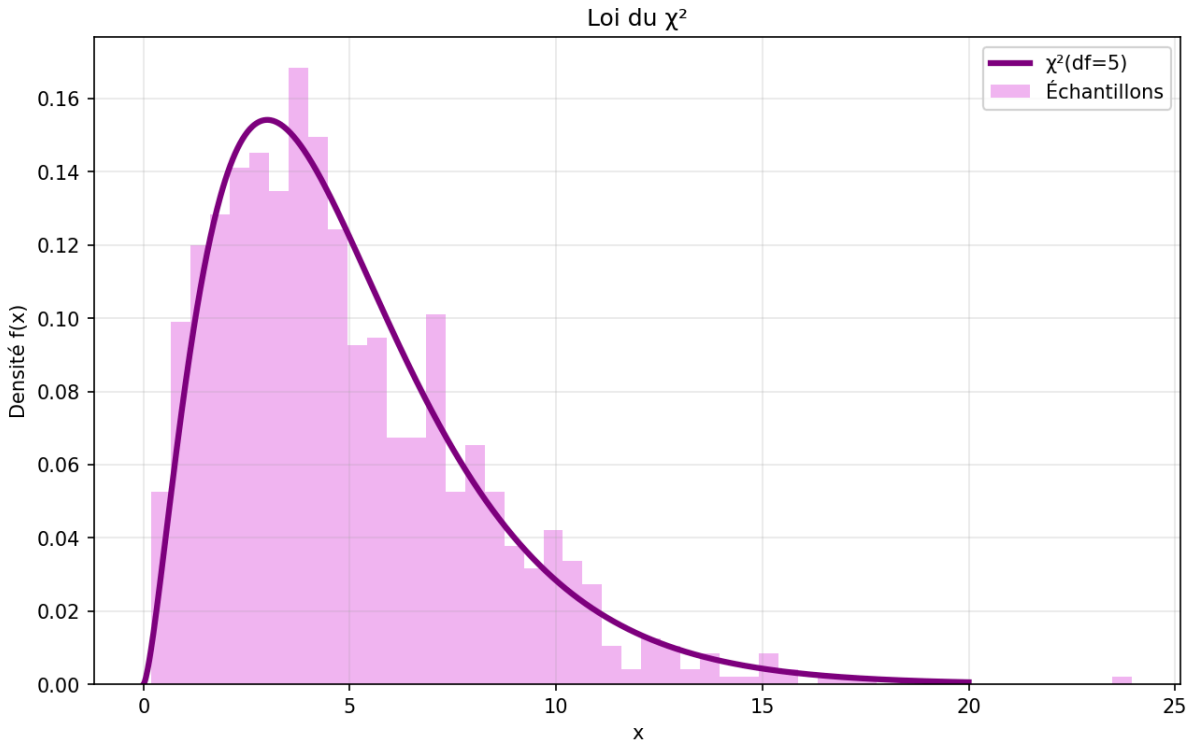
```python
# 5. LOI DU χ² (Histogramme asymétrique + PDF)
def plot_chi2(df=5, size=1000, save="chi2.png"):
    samples = stats.chi2.rvs(df, size=size)
    x = np.linspace(0, 20, 1000)
    pdf = stats.chi2.pdf(x, df)

    plt.figure(figsize=(10,6))
    plt.hist(samples, bins=50, density=True, alpha=0.6, color='violet', label='Échantillons')
    plt.plot(x, pdf, 'purple', lw=3, label=f'χ²(df={df})')
    plt.title('Loi du χ²')
    plt.xlabel('x'); plt.ylabel('Densité f(x)')
    plt.legend(); plt.grid(True, alpha=0.3)
    plt.savefig(f'{IMG_DIR}/{save}', dpi=150, bbox_inches='tight')
    plt.show()
    plt.close()
```

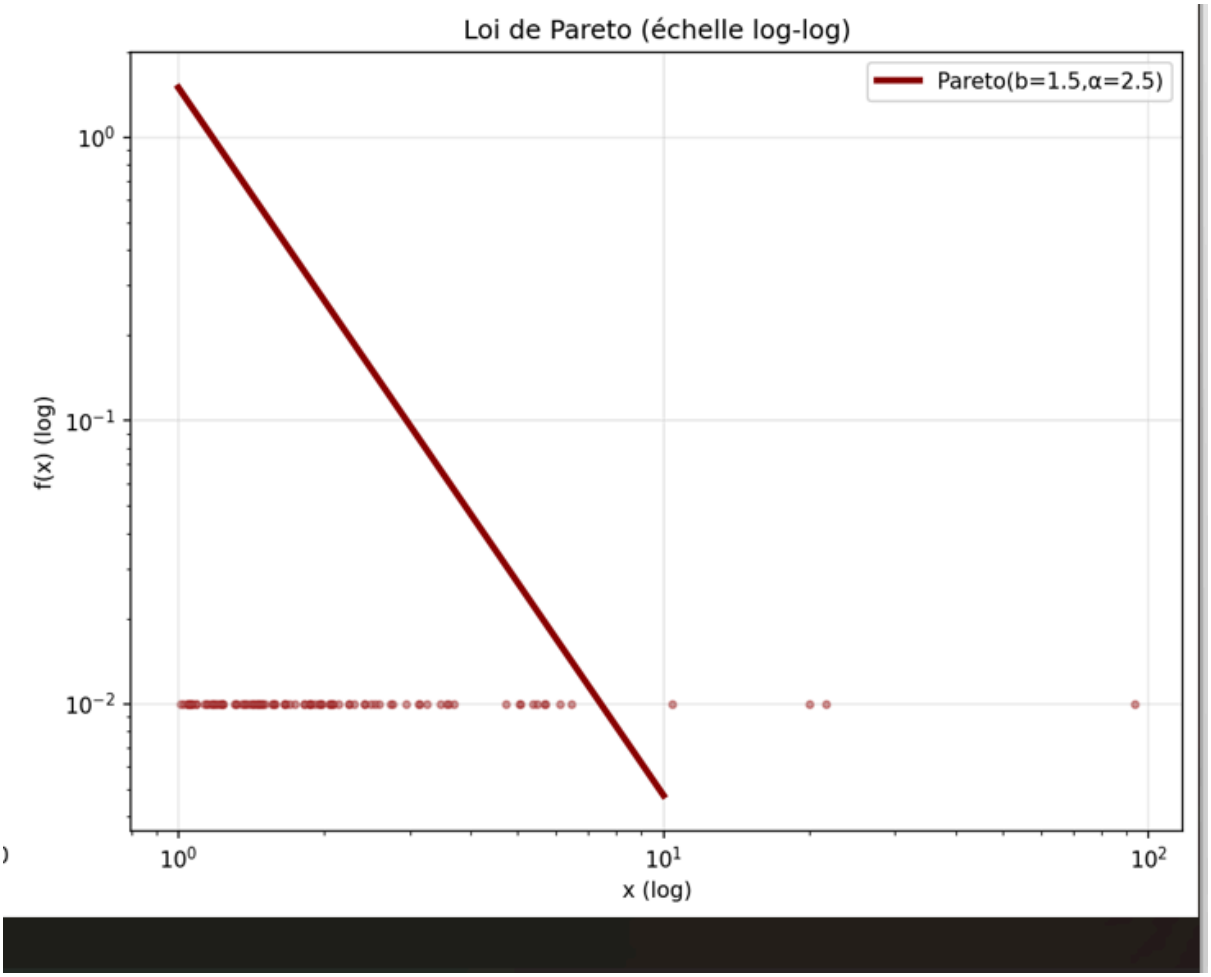
```python
# 6. LOI DE PARETO (Histogramme queue lourde + PDF (échelle log-log))
def plot_pareto(b=1.5, alpha=2.5, size=1000, save="pareto.png"):
    samples = stats.pareto.rvs(b=b, scale=1, size=size)
    x = np.linspace(1, 10, 1000)
    pdf = stats.pareto.pdf(x, b=b, scale=1)

    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15,6))


    # Graphique log-log (caractéristique queue lourde)
    ax2.loglog(x, pdf, 'darkred', lw=3, label=f'Pareto(b={b},α={alpha})')
    ax2.scatter(samples[:100], np.ones(100)*0.01, alpha=0.5, s=10, color='brown')
    ax2.set_title('Loi de Pareto (échelle log-log)')
    ax2.set_xlabel('x (log)'); ax2.set_ylabel('f(x) (log)')
    ax2.legend(); ax2.grid(True, alpha=0.3)

    plt.tight_layout()
    plt.savefig(f'{IMG_DIR}/{save}', dpi=150, bbox_inches='tight')
    plt.show()
    plt.close()

#Affichage
```

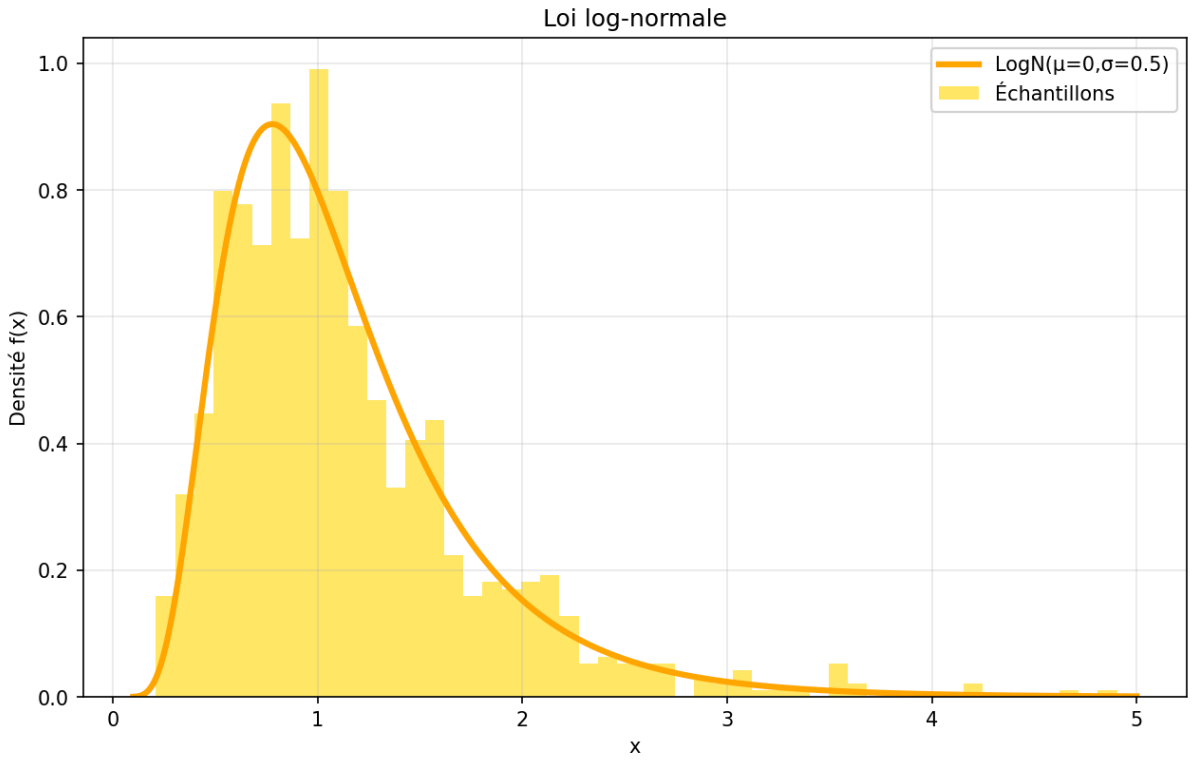
J’utilise le programme suivant pour l’affichage dans le dossier img

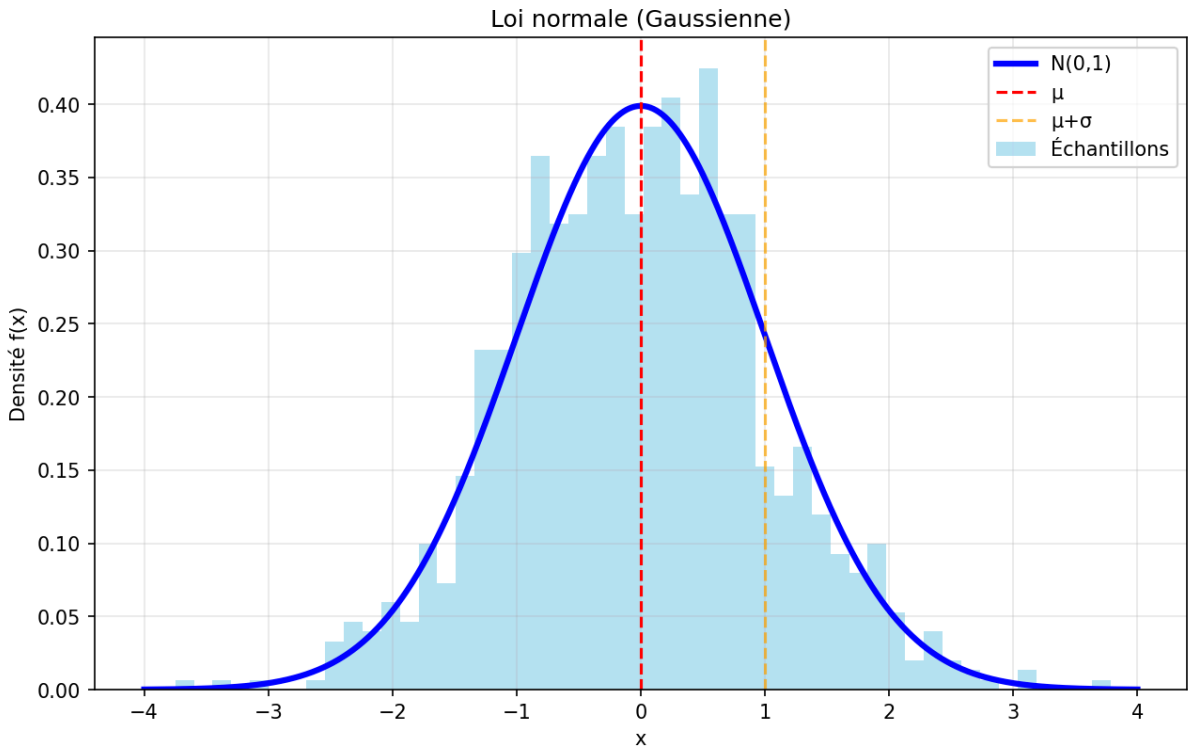
```python
#Affichage
if __name__ == "__main__":
    plot_poisson_continue()
    plot_normale()
    plot_lognormale()
    plot_uniforme()
    plot_chi2()
    plot_pareto()
```

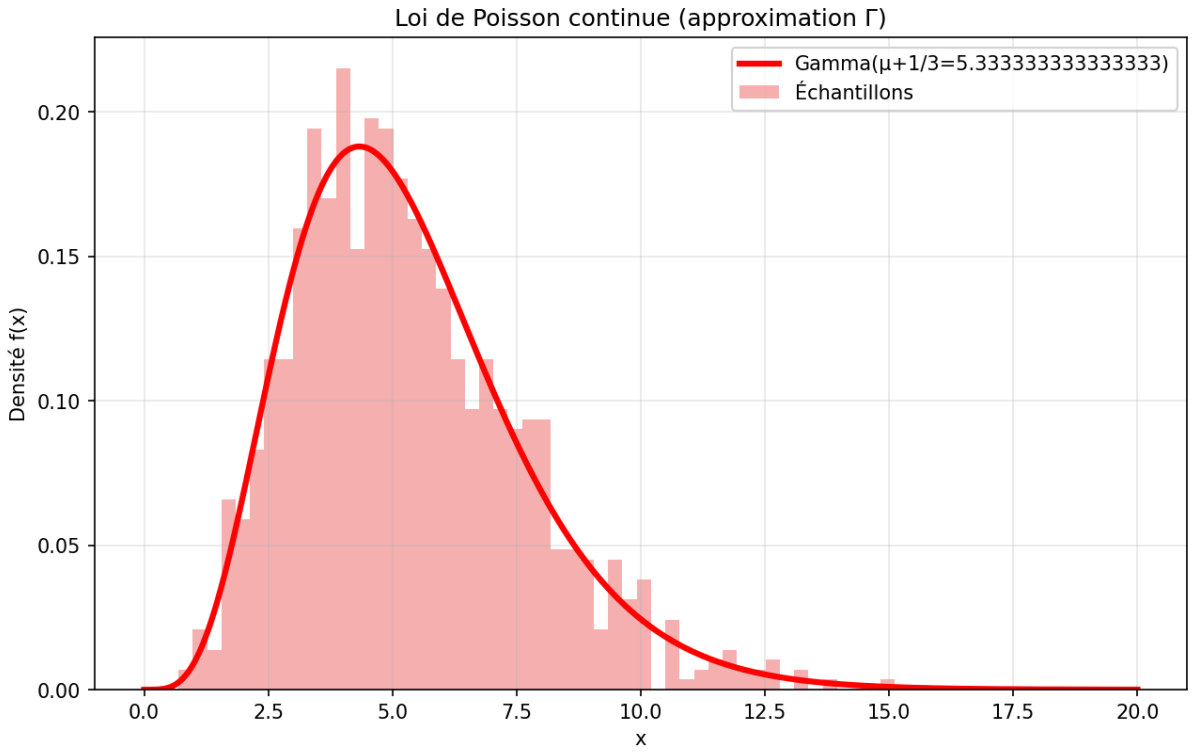
Ce qui donne les résultats suivant:


Loi du χ²
χ²(df=5)
Échantillons
Densité f(x)
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20
25
x


Loi de Pareto (échelle log-log)
Pareto(b=1.5,α=2.5)
f(x) (log)
10^{0}
10^{-1}
10^{-2}
10^{0}
10^{1}
10^{2}
x (log)


Loi log-normale
LogN(μ=0,σ=0.5)
Échantillons
Densité f(x)
x
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5


Loi normale (Gaussienne)
N(0,1)
μ
μ+σ
Échantillons
Densité f(x)
x
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
−4
−3
−2
−1
0
1
2
3
4



- Etape 2:

A l'aide de fonctions informatiques, je dois calculer la moyenne et l'écart-type de chacune de ces distributions, avec les programmes suivants:

```
#Etape 2
#LOI DE DIRAC
def stats_dirac(k0=0):
    print(f" LOI DE DIRAC (k₀={k0})")
    print(f"   Moyenne théorique : {float(k0):.4f}")
    print(f"   Écart-type théorique : {0.0000:.4f}\n")
```

```
#LOI UNIFORME DISCRÈTE
def stats_uniforme_discrete(n=10):
    distrib = stats.randint(0, n)
    print(f" LOI UNIFORME DISCRÈTE [0,{n-1}]")
    print(f"   Moyenne théorique : {distrib.mean():.4f}")
    print(f"   Écart-type théorique : {distrib.std():.4f}\n")
```

```python
#LOI BINOMIALE
def stats_binomiale(n=20, p=0.5):
    distrib = stats.binom(n, p)
    print(f" LOI BINOMIALE B({n},{p})")
    print(f"   Moyenne théorique : {distrib.mean():.4f}")
    print(f"   Écart-type théorique : {distrib.std():.4f}\n")

#LOI DE POISSON (DISCRÈTE)
```

```python
#LOI DE POISSON (DISCRÈTE)
def stats_poisson(mu=5):
    distrib = stats.poisson(mu)
    print(f" LOI DE POISSON (λ={mu})")
    print(f"   Moyenne théorique : {distrib.mean():.4f}")
    print(f"   Écart-type théorique : {distrib.std():.4f}\n")
```

```python
#LOI ZIPF-MANDELBROT
def stats_zipf_mandelbrot(n_max=50, alpha=1.5):
    k = np.arange(1, n_max+1)
    pmf = k ** (-alpha)
    pmf = pmf / pmf.sum()
    mean_zipf = np.sum(k * pmf)
    var_zipf = np.sum((k**2) * pmf) - mean_zipf**2
    std_zipf = np.sqrt(var_zipf)
    print(f" LOI ZIPF-MANDELBROT (α={alpha}, n_max={n_max})")
    print(f"   Moyenne théorique : {mean_zipf:.4f}")
    print(f"   Écart-type théorique : {std_zipf:.4f}\n")
```

```python
#LOI DE POISSON CONTINUE
def stats_poisson_continue(mu=5):
    distrib = stats.gamma(a=mu+1/3, scale=1)
    print(f" LOI DE POISSON CONTINUE Γ(μ+1/3={mu+1/3:.2f})")
    print(f"   Moyenne théorique : {distrib.mean():.4f}")
    print(f"   Écart-type théorique : {distrib.std():.4f}\n")

#LOI NORMALE
def stats_normale(mu=0, sigma=1):
    distrib = stats.norm(mu, sigma)
    print(f" LOI NORMALE N({mu},{sigma})")
    print(f"   Moyenne théorique : {distrib.mean():.4f}")
    print(f"   Écart-type théorique : {distrib.std():.4f}\n")

#LOI LOG-NORMALE
def stats_lognormale(mu=0, sigma=0.5):
    distrib = stats.lognorm(s=sigma, scale=np.exp(mu))
    print(f" LOI LOG-NORMALE (μ={mu},σ={sigma})")
    print(f"   Moyenne théorique : {distrib.mean():.4f}")
    print(f"   Écart-type théorique : {distrib.std():.4f}\n")

#LOI UNIFORME CONTINUE
def stats_uniforme_continue(a=0, b=1):
    distrib = stats.uniform(a, b-a)
    print(f" LOI UNIFORME CONTINUE [{a},{b}]")
    print(f"   Moyenne théorique : {distrib.mean():.4f}")
    print(f"   Écart-type théorique : {distrib.std():.4f}\n")
```

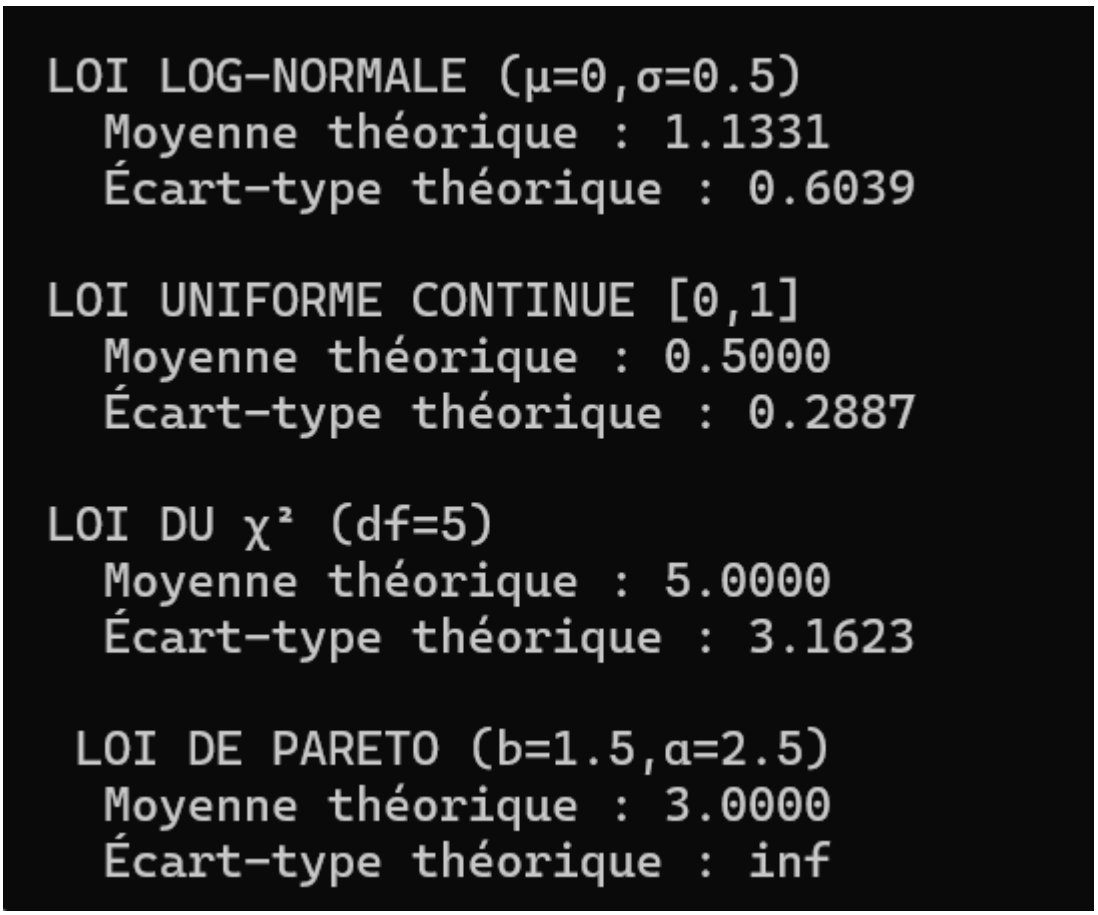
```python
#LOI DU χ²
def stats_chi2(df=5):
    distrib = stats.chi2(df)
    print(f" LOI DU χ² (df={df})")
    print(f"   Moyenne théorique : {distrib.mean():.4f}")
    print(f"   Écart-type théorique : {distrib.std():.4f}\n")

#LOI DE PARETO
def stats_pareto(b=1.5, alpha=2.5):
    distrib = stats.pareto(b, scale=1)
    print(f"  LOI DE PARETO (b={b},α={alpha})")
    print(f"   Moyenne théorique : {distrib.mean():.4f}")
    print(f"   Écart-type théorique : {distrib.std():.4f}\n")
```

Pour afficher les résultats, je mets le programme:

```python
if __name__ == "__main__":
    print("=== MOYENNE ET ÉCART-TYPE THÉORIQUES ===\n")

    # Discrètes (5)
    stats_dirac()
    stats_uniforme_discrete()
    stats_binomiale()
    stats_poisson()
    stats_zipf_mandelbrot()

    # Continues (6)
    stats_poisson_continue()
    stats_normale()
    stats_lognormale()
    stats_uniforme_continue()
    stats_chi2()
    stats_pareto()

    print()
```

Ce qui affiche les résultats suivants sur le terminal:

```
randint', 'zipf', 'zipfian']
=== MOYENNE ET ÉCART-TYPE THÉORIQUES ===

LOI DE DIRAC (k₀=0)
  Moyenne théorique : 0.0000
  Écart-type théorique : 0.0000

LOI UNIFORME DISCRÈTE [0,9]
  Moyenne théorique : 4.5000
  Écart-type théorique : 2.8723

LOI BINOMIALE B(20,0.5)
  Moyenne théorique : 10.0000
  Écart-type théorique : 2.2361

LOI DE POISSON (λ=5)
  Moyenne théorique : 5.0000
  Écart-type théorique : 2.2361

LOI ZIPF-MANDELBROT (α=1.5, n_max=50)
  Moyenne théorique : 5.4709
  Écart-type théorique : 8.5216

LOI DE POISSON CONTINUE Γ(μ+1/3=5.33)
  Moyenne théorique : 5.3333
  Écart-type théorique : 2.3094

LOI NORMALE N(0,1)
  Moyenne théorique : 0.0000
  Écart-type théorique : 1.0000
```



Les résultats affichés sont théoriques, c'est-à-dire sont les formules mathématiques exactes, ce qui semblait plus cohérent avec l'exercice plutôt que d'afficher les résultats empiriques avec les échantillons utilisés à l'étape 1.

SÉANCE 5:

I-Questions de cours

1) **Comment définir l'échantillonnage. Pourquoi ne pas utiliser la population en entier? Quelles sont les méthodes d'échantillonnage? Comment les choisir ?**

L’échantillonnage représente le processus de sélection d’une sous-partie représentative d’une population globale pour en inférer des propriétés sur l’ensemble. Recourir à la totalité de la population s’avère souvent impraticable en raison de contraintes financières, temporelles, logistiques ou de l’immensité même de cette population, rendant une étude exhaustive matériellement impossible ou excessivement onéreuse.

- Méthodes aléatoires

Ces approches, comme le tirage au sort simple (avec ou sans remise), l’échantillonnage stratifié ou systématique, assurent une représentativité via le hasard et autorisent l’application précise des lois statistiques pour contrôler l’erreur d’échantillonnage.

- Méthodes non aléatoires

Elles incluent les quotas, le systématique ou les techniques Monte Carlo, utiles lorsque la base de données est incomplète ou pour pallier des limites pratiques.

La décision repose sur la structure de la population, les objectifs de l’étude, le budget disponible, la taille globale, la précision requise et les contraintes d’accessibilité.

2) Comment définir un estimateur et une estimation?

Un estimateur constitue une fonction mathématique appliquée aux données observées d’un échantillon pour évaluer un paramètre inconnu de la population, tel que la moyenne, la variance ou une proportion. L’estimation désigne la valeur numérique concrète résultant de cette opération sur l’échantillon.
L’estimateur représente l’outil ou la règle de calcul tandis que l’estimation est le résultat spécifique obtenu.

3) Comment distingueriez-vous l’intervalle de fluctuation et l’intervalle de confiance ?

L’intervalle de fluctuation mesure la variabilité attendue d’une proportion observée lors de multiples échantillonnages répétés, en partant d’un paramètre théorique connu comme *p*. Il reflète les oscillations dues au hasard dans le processus d’échantillonnage lui-même.
L'intervalle de confiance encadre un paramètre inconnu à partir d’un unique échantillon, en associant un niveau de certitude précis, et sert ainsi d’instrument d’estimation.

Le premier s’appuie sur un paramètre connu pour décrire le comportement de fréquences multiples ; le second, construit depuis des données uniques, vise à borner la valeur réelle d’un paramètre incertain.

4) **Qu’est-ce qu’un biais dans la théorie de l’estimation?**

Un biais correspond à l’écart systématique entre l’espérance mathématique d’un estimateur et la vraie valeur du paramètre de la population. Un estimateur biaisé surestime ou sous-estime constamment ce paramètre, y compris pour des échantillons très volumineux.

5) **Comment appelle-t-on une statistique travaillant sur la population totale? Faites le lien avec la notion de données massives 1 ?**

La statistique exhaustive traite l'intégralité de la population, mesurant tous les paramètres sans inférence ni échantillonnage.
Les big data tendent vers cette exhaustivité par leur volume géant, où l'analyse descriptive remplace souvent l'inférence, rendant l'échantillonnage inutile ; la qualité et la représentativité persistent comme priorités.

6) **Quels sont les enjeux autour du choix d’un estimateur ?**

Le choix d’un estimateur repose sur un arbitrage les enjeux suivants :
- une absence de biais pour une estimation fidèle au paramètre réel.
- une variance faible pour une précision élevée et des résultats stables.
- une résistance aux valeurs aberrantes ou extrêmes pour une robustesse accrue.
- une simplicité de mise en œuvre pour une application pratique.
- une minimisation de l’erreur quadratique moyenne pour une efficacité optimale.
- une optimalité via les théorèmes de Rao-Blackwell et Lehmann-Scheffé, réservée aux estimateurs sans biais à variance minimale, adaptés au contexte empirique.

7) **Quelles sont les méthodes d’estimation d’un paramètre? Comment en sélectionner une ?**

- L’estimation ponctuelle : elle associe au paramètre une seule valeur estimée à partir de l’échantillon, comme la moyenne ou la proportion observée.
- L’estimation par intervalle : elle fournit une plage de valeurs, appelée intervalle de confiance, dans laquelle le paramètre a une forte probabilité de se situer.
- L’estimation par maximum de vraisemblance : elle consiste à choisir la valeur du paramètre qui rend les données observées les plus probables.
- La méthode des moments : elle repose sur l’égalité entre les moments théoriques et empiriques de la distribution.
- L’approche bayésienne : elle incorpore une connaissance a priori sur le paramètre pour actualiser l’estimation à partir des données disponibles.

- Les méthodes non paramétriques (bootstrap) : elles ne reposent pas sur d'hypothèses fortes concernant la distribution de la population et utilisent des rééchantillonnages pour estimer le paramètre.

Le choix de la méthode dépend de la nature des données, des hypothèses que l'on accepte de formuler sur la population ou le modèle probabiliste, du type de paramètre à estimer et de la précision souhaitée.

8) Quels sont les tests statistiques existants ? À quoi servent-ils ? Comment créer un test ?

Les tests statistiques servent à vérifier la validité d'une hypothèse portant sur un paramètre d'une population. Ils permettent de décider, en tenant compte d'un risque d'erreur (risque alpha), si une hypothèse doit être acceptée ou rejetée.
On distingue principalement deux grandes catégories :
- Les tests paramétriques, tels que le test de Student, le test du Khi² ou le test de Fisher, qui reposent sur des hypothèses concernant la distribution (généralement normale) de la population.
- Les tests non paramétriques, comme ceux de Mann–Whitney ou de Kolmogorov–Smirnov, utilisés lorsque la loi de distribution n'est pas connue ou que les conditions d'application des tests paramétriques ne sont pas respectées.
- Parmi les tests spécifiques, on trouve également le test de normalité (Shapiro–Wilk) qui sert à vérifier si les données suivent une distribution normale.

La création d'un test statistique suit plusieurs étapes :
1) Formuler les hypothèses nulle (H_0) et alternative (H_1).
2) Choisir une statistique de test adaptée au problème.
3) Déterminer le seuil de signification α (risque d'erreur toléré).
4) Calculer la statistique à partir des données et la comparer à la loi de référence pour décider d'accepter ou de rejeter H_0.

9) Que pensez-vous des critiques de la statistique inférentielle ?

Je pense que les critiques faites à la statistique inférentielle sont largement justifiées, surtout quand on voit à quel point certains usages en déforment le sens. On s'appuie trop souvent de manière automatique sur le seuil de 5% sans réfléchir à ce qu'il représente vraiment, et on oublie que les conditions théoriques — comme la normalité ou l'homogénéité des variances — sont rarement parfaitement respectées. Ces simplifications peuvent mener à des conclusions trompeuses, d'autant plus avec les dérives comme le "p-hacking" ou l'interprétation erronée des "p-values".

Pour autant, je ne crois pas qu'il faille rejeter la statistique inférentielle. Si elle est utilisée avec rigueur, en tenant compte du contexte et des limites de ses hypothèses, elle reste un outil essentiel pour quantifier l'incertitude et étayer des conclusions à

partir d'échantillons. Même à l'ère du big data, elle nous oblige à penser la généralisation et la reproductibilité des résultats. En somme, ce ne sont pas les principes de l'inférence qui posent problème, mais la manière dont on les applique.

II- Mise en pratique

SÉANCE 6:

I-Questions de cours:

1) **Qu'est-ce qu'une statistique ordinale ? À quelle autre statistique catégorielle s'oppose-telle ? Quel type de variables utilise-t-elle ? En quoi cela peut matérialiser une hiérarchie spatiale ?**

Une statistique ordinale correspond à l'ensemble des méthodes fondées sur le classement d'objets ou d'individus selon un ordre, plutôt que sur leurs valeurs absolues. Elle repose sur les rangs obtenus à partir d'une série d'observations ordonnées. Elle s'oppose ainsi à la statistique nominale, qui regroupe les individus dans des catégories sans lien hiérarchique ni ordre interne.

Ce type de statistique mobilise des "variables ordinales", c'est-à-dire des variables qualitatives dont les modalités peuvent être classées selon un ordre croissant ou décroissant. Ces variables traduisent souvent une échelle de mesure graduée, comme un niveau de risque, d'intensité ou de taille.

En géographie, la statistique ordinale est particulièrement utile pour mettre en évidence une "hiérarchie spatiale". En classant des entités — par exemple des villes selon leur population ou des territoires selon leur richesse — elle permet de représenter les positions relatives dans l'espace. Cette hiérarchisation rend visibles les rapports d'ordre, les structures d'inégalités ou les dynamiques de domination au sein d'un système spatial donné, transformant l'espace en un ensemble organisé et interprétable.

2) **Quel ordre est à privilégier dans les classifications ?**

Dans une classification ordinale, l'ordre décroissant est à privilégier car il rend immédiatement visibles les dominances et les hiérarchies, et même si un tri croissant reste possible, il est moins intuitif pour représenter un espace hiérarchisé. Placer les valeurs les plus fortes en tête de liste permet de lire directement les rapports de grandeur.

3) **Quelle est la différence entre une corrélation des rangs et une concordance de classements ?**

La corrélation des rangs et la concordance des classements sont deux approches ordinales permettant de comparer des ordres, mais elles diffèrent par leur logique d'évaluation.

- La corrélation des rangs, mesurée notamment par les coefficients de Spearman ou de Kendall, évalue la proximité globale entre deux séries ordonnées. Elle indique si, dans l'ensemble, les rangs élevés pour une variable correspondent à des rangs élevés pour une autre, traduisant ainsi une relation monotone entre les deux distributions.
- La concordance des classements, en revanche, s'intéresse à la cohérence paire par paire entre les ordres établis. Elle repose sur le comptage des paires concordantes et discordantes : la concordance est dite complète lorsque toutes les paires vont dans le même sens, et nulle lorsque le nombre de concordances et de discordances s'équilibre. Cette méthode analyse donc la structure interne des classements plutôt que leur tendance générale.

Ainsi, la corrélation des rangs mesure une similarité globale entre deux ordres, tandis que la concordance des classements examine plus finement l'accord structurel entre les positions relatives des individus dans chacun d'eux.

4) **Quelle est la différence entre les tests de Spearman et de Kendall ?**

Les tests de Spearman et de Kendall reposent tous deux sur la comparaison des rangs mais diffèrent par leur logique de calcul et leur sensibilité aux variations d'ordre.

- Le test de Spearman évalue la force d'une relation monotone entre deux séries ordonnées en comparant directement les rangs par leurs différences – souvent à travers la somme des écarts au carré. Il mesure ainsi une tendance globale et quantitative entre deux classements, ce qui le rend sensible aux valeurs extrêmes mais moins attentif aux inversions locales.
- Le test de Kendall, quant à lui, s'appuie sur le dénombrement des paires concordantes et discordantes. Il évalue combien de fois deux objets sont ordonnés dans le même sens dans les deux séries, traduisant un accord de structure dans les ordres. Ce test est plus robuste aux relations non linéaires et se généralise aisément à plusieurs classements.

Pour ainsi dire, Spearman mesure la proximité globale des rangs de manière quantitative, tandis que Kendall mesure la cohérence qualitative de l'ordre des paires.

5) À quoi servent les coefficients de Goodman-Kruskal et de Yule ?

- Le coefficient de Goodman-Kruskal mesure la force d'association d'ordre entre deux variables ordinales en comparant le nombre de paires concordantes (Na) et discordantes (Nd). Il varie entre -1 et +1 : une valeur de +1 traduit une concordance parfaite, -1 une inversion totale et 0 une absence d'association observable. Proche conceptuellement du τ de Kendall, il indique la cohérence d'ordre entre deux séries tout en exprimant le gain de prévisibilité d'une variable à partir d'une autre dans un tableau de contingence. Il constitue ainsi un indicateur pertinent de la relation d'ordre et de la capacité explicative entre deux variables catégorielles.

- Le coefficient de Yule, ou Q de Yule, est un cas particulier du coefficient de Goodman-Kruskal, réservé aux tableaux de contingence 2×2. Il mesure l'association entre deux variables dichotomiques, comme oui/non ou présent/absent. Il varie également entre -1 et +1 : +1 indique une association positive parfaite, -1 une association négative parfaite et 0 une absence de lien. Ce coefficient fournit une mesure simple et directe de la dépendance entre deux modalités opposées et constitue un outil adapté à l'analyse de relations binaires.

II- Mise en pratique