

# **Cas d'Utilisation des Graphiques avec Matplotlib et Seaborn**

**Professeur: Mr Youssouf Vessou TRAORE**

**Présenté par : Abdoulaye CAMARA & Ibrahima Bah**

# Table des matières

1. Introduction à Matplotlib et Seaborn
2. Objectif de la présentation :
3. Pourquoi Visualiser les Données
4. Études comparatives des différents types de diagrammes.
5. Conclusion et Recommandations

# 1. Introduction à Matplotlib et Seaborn

## a. Matplotlib : Le Couteau Suisse de la Visualisation

Matplotlib est une bibliothèque fondamentale en Python, offrant un contrôle granulaire sur chaque aspect d'un graphique. Elle est idéale pour créer des visualisations hautement personnalisées, des figures complexes et des graphiques prêts pour la publication scientifique. Sa flexibilité permet de construire des graphiques à partir de zéro, en définissant précisément les axes, les titres, les légendes, les couleurs et les styles.

### Points clés :

- 1. Grande flexibilité et contrôle total.**
- 2. Permet des visualisations très personnalisées.**
- 3. Base pour de nombreuses autres bibliothèques.**

## b. Seaborn : L'Esthétique Statistique Simplifiée

Seaborn est une bibliothèque de visualisation de données basée sur Matplotlib. Elle excelle dans la création de visualisations statistiques attrayantes avec moins de code. Seaborn est particulièrement utile pour l'analyse exploratoire de données, car elle intègre des fonctionnalités pour gérer les DataFrames Pandas et automatise de nombreux aspects esthétiques, rendant les graphiques plus lisibles et plus beaux par défaut.

### Points clés :

- Simplifie les visualisations statistiques complexes.**
- Esthétiques par défaut soignées.**
- Excellente intégration avec les DataFrames Pandas.**

## 2. Objectif de la présentation :

L'objectif de cette présentation est d'analyser de manière critique les bibliothèques Matplotlib et Seaborn, couramment utilisées en Python pour la visualisation de données. Il s'agira d'identifier leurs points forts et leurs limites respectives, tant sur le plan technique qu'en termes d'usages pédagogiques et professionnels.

Cette étude vise à fournir des éléments de réflexion permettant de choisir la bibliothèque la plus appropriée selon les objectifs visés, la nature des données, ainsi que le niveau de personnalisation ou de complexité requis pour les représentations graphiques.

## 3. Pourquoi Visualiser les Données ?

La visualisation des données est bien plus qu'une simple présentation esthétique ; c'est un outil essentiel pour l'analyse exploratoire de données (EDA), la détection de tendances et la communication efficace des résultats. Dans un monde inondé d'informations, les données brutes, même structurées, peuvent être accablantes et difficiles à interpréter sans une représentation visuelle adéquate.

**Les graphiques transforment des données brutes en insights exploitables. Ils permettent aux data scientists et aux analystes de :**

- **Identifier des motifs et des anomalies :** Un simple graphique peut révéler des corrélations, des clusters ou des valeurs aberrantes qui seraient invisibles dans un tableau numérique.
- **Valider des hypothèses :** Visualiser les relations entre les variables aide à confirmer ou infirmer des suppositions initiales sur le dataset.
- **Communiquer des découvertes :** Un graphique bien conçu est un langage universel qui permet de présenter des résultats complexes de manière concise et mémorable à un public non technique.

- **Guider la prise de décision :** Les décideurs peuvent s'appuyer sur des visualisations claires pour comprendre rapidement les implications des données et agir en conséquence.

## 4. Études comparatives des différents types de diagrammes.

### 1. Graphique de Dispersion (Scatter Plot)

Le graphique de dispersion est un outil fondamental pour visualiser la relation entre deux variables continues. Il permet d'identifier des tendances, des corrélations (positives, négatives, nulles) et la présence d'outliers (valeurs aberrantes). Chaque point représente une observation, avec sa position sur les axes X et Y déterminée par les valeurs des deux variables.

#### Cas d'utilisation :

- Explorer la relation entre deux variables numériques continues.
- Détecter des clusters ou des schémas de regroupement.
- Identifier des valeurs extrêmes ou aberrantes.

#### Exemple concret :

Analyse de la relation entre l'âge d'un employé et son salaire annuel au sein d'une entreprise. On pourrait aussi ajouter une troisième dimension via la couleur (par exemple, le département) pour voir si des tendances salariales diffèrent par département.

**Comparaison :** Matplotlib offre un contrôle plus granulaire pour des personnalisations spécifiques (ex. : taille de point variable, flèches d'annotation). Seaborn excelle dans la simplification des analyses multi-variables (couleur par catégorie, taille par valeur) et l'intégration facile de lignes de régression, produisant des graphiques plus esthétiques avec moins d'effort.

## 2. Graphique en Lignes (Line Plot)

Le graphique en lignes est idéal pour visualiser l'évolution d'une ou plusieurs variables continues sur une période donnée ou un ordre séquentiel. Il met en évidence les tendances, les cycles, les pics et les creux.

### Cas d'utilisation :

- Afficher des séries temporelles (ex. : prix des actions, température).
- Comparer l'évolution de plusieurs catégories ou groupes au fil du temps.
- Identifier des motifs cycliques ou des changements brusques.

**Exemple concret :** Visualisation de l'évolution du nombre d'utilisateurs actifs mensuels (MAU) pour une application mobile sur les 12 derniers mois, avec une distinction entre utilisateurs iOS et Android.

**Comparaison :** Matplotlib offre un contrôle total pour les tracés ligne par ligne, idéal pour des séries temporelles avec des spécificités uniques. Seaborn, avec lineplot, est plus intuitif pour visualiser l'évolution de multiples groupes issus d'un DataFrame, calculant les agrégats et les intervalles de confiance par défaut.

## 3. Histogramme

L'histogramme est utilisé pour visualiser la distribution d'une variable numérique continue. Il divise la plage de valeurs de la variable en intervalles (bins) et compte le nombre d'observations dans chaque intervalle, affichant ces comptes sous forme de barres. Il permet de comprendre la forme de la distribution (normale, asymétrique, multimodale), sa centralité et sa dispersion.

### Cas d'utilisation :

- Analyser la distribution d'une seule variable continue.
- Identifier des motifs cycliques ou des changements brusques.

- Détecter la skewness (asymétrie) ou la présence de multiples modes.

**Exemple concret :** Distribution des âges des clients dans une base de données, pour déterminer si la clientèle est majoritairement jeune, âgée, ou uniformément répartie.

**Comparaison :** Matplotlib est parfait pour un contrôle précis sur les bins et l'apparence brute de l'histogramme. Seaborn enrichit l'histogramme avec des fonctionnalités statistiques comme l'estimation de densité de noyau (KDE) et la facilité de superposition de distributions pour différentes catégories, le rendant idéal pour l'EDA.

#### 4. Boîte à Moustaches (Box Plot)

Le box plot (ou diagramme en boîtes) est un moyen standardisé de représenter la distribution d'une variable numérique et de comparer les distributions entre différentes catégories. Il affiche les quartiles (Q1, médiane, Q3), les valeurs minimales/maximales (sans les outliers) et les outliers individuels.

##### **Cas d'utilisation :**

- Visualiser la distribution d'une variable numérique.
- Comparer les distributions entre plusieurs groupes ou catégories.
- Identifier la présence d'outliers et évaluer la dispersion.

**Exemple concret :** Comparaison des salaires médians et de la dispersion salariale entre différents départements d'une entreprise (par exemple, IT, Ventes, RH). Cela permet de voir rapidement où les salaires sont les plus élevés, où la variabilité est la plus grande, et s'il existe des outliers.

**Comparaison :** Matplotlib offre un contrôle plus direct pour la personnalisation de chaque élément du box plot. Seaborn, cependant, est plus performant pour la comparaison rapide de distributions entre plusieurs catégories et peut être combiné avec d'autres types de tracés

pour une visualisation plus riche (ex. : swarmplot ou stripplot superposés).

## 5. Diagramme en Barres (Bar Plot)

Les diagrammes en barres sont utilisés pour représenter et comparer les quantités ou fréquences de différentes catégories. La longueur de chaque barre est proportionnelle à la valeur qu'elle représente. Ils sont particulièrement efficaces pour des comparaisons claires entre des groupes discrets.

### Cas d'utilisation :

- Comparer des valeurs entre différentes catégories.
- Afficher des fréquences ou des comptes pour des données catégorielles.
- Visualiser des totaux ou des moyennes par groupe.

**Exemple concret :** Nombre de ventes réalisées par chaque région (Nord, Sud, Est, Ouest) au cours d'un trimestre. On peut aussi empiler des barres pour montrer les ventes par produit dans chaque région.

**Comparaison :** Matplotlib est préférable pour des ensembles de données plus petits ou quand une personnalisation très spécifique de chaque barre est requise. Seaborn excelle avec les DataFrames Pandas, fournissant des fonctions pour des bar plots statistiques complexes (moyennes, erreurs standard, groupements) avec des esthétiques intégrées.

## 6. Carte de Chaleur (Heatmap) et Diagramme en Violon (Violin Plot)

Ces deux types de graphiques offrent des perspectives uniques sur les distributions et les corrélations de données, chacun ayant ses forces spécifiques pour des cas d'utilisation distincts.

### 1. Carte de Chaleur (Heatmap) :

Une carte de chaleur est une représentation graphique de données où les valeurs individuelles d'une matrice sont représentées sous forme de



couleurs. Elle est particulièrement utile pour visualiser des matrices de corrélation,

l'importance des caractéristiques, ou des données tabulaires complexes.

**Cas d'utilisation :** Visualiser des matrices de corrélation entre variables, l'activité d'un utilisateur sur un calendrier, ou des données tabulaires avec de nombreuses colonnes et lignes.

**Exemple concret :** Matrice de corrélation des caractéristiques d'un dataset pour identifier les relations entre elles.

## 2. Diagramme en Violon (Violin Plot)

Le diagramme en violon combine un box plot avec une estimation de la densité de noyau (KDE) de la distribution des données. Il montre non seulement les statistiques résumées (médiane, quartiles) mais aussi la forme de la distribution, offrant une vue plus riche que le box plot seul.

**Cas d'utilisation :** Comparer la distribution d'une variable numérique à travers différentes catégories, particulièrement lorsque la forme de la distribution est importante (ex. : multimodale).

**Exemple concret :** Comparer la distribution des prix de logements par quartier, pour voir non seulement le prix médian mais aussi la concentration des prix dans chaque quartier.

**Comparaison :** Matplotlib peut techniquement créer ces graphiques, mais Seaborn simplifie grandement leur implémentation et fournit des esthétiques supérieures par défaut. Pour les heatmaps, Seaborn est le choix préféré. Pour les diagrammes en violon, qui sont une amélioration des box plots en termes d'information de densité, Seaborn est quasi-exclusif par sa facilité d'utilisation et ses visualisations riches.

## 5. Conclusion et Recommandations

À travers cette présentation, nous avons exploré les cas d'utilisation spécifiques des graphiques avec Matplotlib et Seaborn, deux bibliothèques incontournables pour la visualisation de données en Python. Il est clair que loin d'être des concurrents, elles sont complémentaires et peuvent même être utilisées de concert pour des résultats optimaux.

### 1. Matplotlib

- **Flexibilité et contrôle granulaire** : Chaque élément d'un graphique peut être ajusté.
- **Idéal pour graphiques complexes/scientifiques** : Quand la précision et la personnalisation sont primordiales.
- **Base de nombreuses bibliothèques** : Comprendre Matplotlib aide à maîtriser d'autres outils.

### 2. Seaborn

- **Simplicité et esthétiques par défaut** : Moins de code pour des graphiques attrayants.
- **Intégration avec Pandas** : Excellente pour l'analyse exploratoire de données.
- **Spécialisation en statistiques** : Fonctions dédiées pour des visualisations statistiques avancées.

### 3. Quand choisir l'un ou l'autre ?

**Seaborn pour l'analyse exploratoire rapide** : Lorsque vous avez besoin de comprendre rapidement les relations et distributions dans vos données, et que vous privilégiez des graphiques clairs et esthétiques avec un minimum d'effort.

- **Matplotlib pour la personnalisation et la publication :** Quand vous avez des exigences spécifiques en matière de design, de mise en page, ou lorsque vous préparez des figures pour des publications scientifiques ou des rapports finaux où chaque détail doit être précis.
- **Combinaison des deux :** Une approche courante est d'utiliser Seaborn pour générer les graphiques de base avec leurs esthétiques supérieures, puis d'utiliser Matplotlib pour des ajustements finaux comme l'ajout de titres personnalisés, d'annotations ou l'ajustement des limites d'axes.

**Invitation à expérimenter :** Le meilleur moyen de maîtriser ces bibliothèques est de les utiliser. N'hésitez pas à télécharger des datasets publics (comme ceux disponibles via `sns.load_dataset()`) et à expérimenter avec différents types de graphiques pour voir comment ils révèlent les patterns cachés dans vos données.