



Programmation Python

Séance 8 — Visualiser et faire le pont

Dr. El Hadji Bassirou TOURÉ

Département de Mathématiques et Informatique, Faculté des Sciences et Techniques, Université Cheikh Anta Diop de Dakar

<https://ebtoure.sn/cours/programmation-python/chapitres/08-visualiser-et-faire-le-pont.html>

Version du 11 octobre 2026



Version en ligne

Programmation Python

— Séance 8

Visualiser et faire le pont

Dr. El Hadji Bassirou TOURÉ

Département de Mathématiques et Informatique, Faculté des Sciences et Techniques, Université Cheikh Anta Diop de Dakar

Objectifs de la séance

De quoi parle cette séance

Jusqu'ici, on a appris à **calculer** sur des données. Aujourd'hui, on apprend à les **regarder** — et à comprendre ce qu'un graphique dit, ou ne dit pas. C'est la dernière étape avant le Machine Learning.

- **Pourquoi** visualiser : ce qu'une moyenne ne montre pas.
- **Comment lire** un graphique : ses éléments, ce qu'on y cherche.
- **Lequel choisir** : histogramme, barres ou nuage de points.
- **Le pont vers le ML** : reconnaître un fruit à partir de ses mesures.

Le jeu de données : 30 fruits du marché

Trois fruits, trois mesures

On a pesé et mesuré 10 **mangues**, 10 **bananes** et 10 **oranges**. Pour chaque fruit : sa **masse** (g), sa **longueur** (cm) et son **taux de sucre** (%).

	fruit	masse	longueur	sucré
0	mangue	301	10.3	14.7
1	banane	137	18.4	11.1
2	orange	201	8.1	9.2

Une ligne = un fruit. Une colonne = une mesure. C'est la forme de toute table de données, et celle que le Machine Learning attend.

Pourquoi visualiser

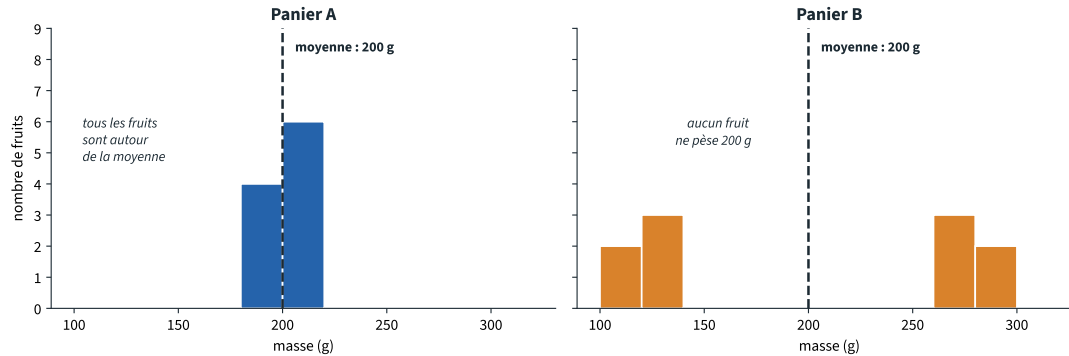
Le problème : une moyenne ne dit pas tout

Une question simple

Deux paniers de fruits ont **exactement la même masse moyenne** : 200 g. Contiennent-ils la même chose ?

Avec les seuls chiffres, impossible de répondre : la moyenne est identique. Il faut **regarder la répartition** — c'est-à-dire compter combien de fruits se trouvent dans chaque tranche de masse.

La réponse, en un coup d'œil



Deux paniers de masse moyenne identique (200 g), mais de composition totalement différente.

Ce que cette figure nous apprend

- **Panier A** : tous les fruits pèsent autour de 200 g. La moyenne **représente bien** le panier — ce sont sans doute dix oranges.
- **Panier B** : aucun fruit ne pèse 200 g ! Il y a des fruits légers (≈ 120 g) et des fruits lourds (≈ 280 g). La moyenne ne correspond à **aucun** fruit réel.

À retenir

Une moyenne résume, mais elle **cache la forme** des données. Deux jeux très différents peuvent avoir la même moyenne : **il faut regarder avant de conclure**.

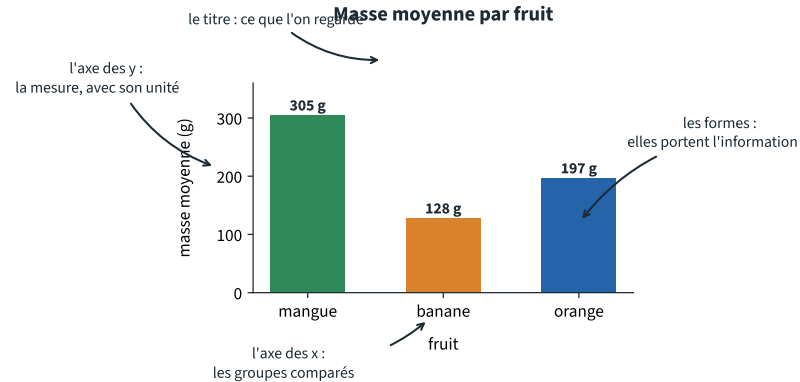
Ce qu'un graphique permet de voir

- **La répartition** : les valeurs sont-elles groupées, étalées, ou séparées en plusieurs paquets ?
- **Les groupes** : y a-t-il des familles distinctes dans les données ?
- **Les liens** : quand une mesure augmente, qu'arrive-t-il à une autre ?
- **Les anomalies** : une valeur aberrante saute aux yeux sur un graphique, jamais dans un tableau.

C'est pourquoi visualiser est le **premier réflexe** de tout travail sur des données — avant le moindre calcul savant.

Lire un graphique

Les éléments d'un graphique



Tout graphique se lit de la même façon : un titre, deux axes légendés, et des formes qui portent l'information.

Un graphique sans légende ne veut rien dire

L'erreur la plus fréquente

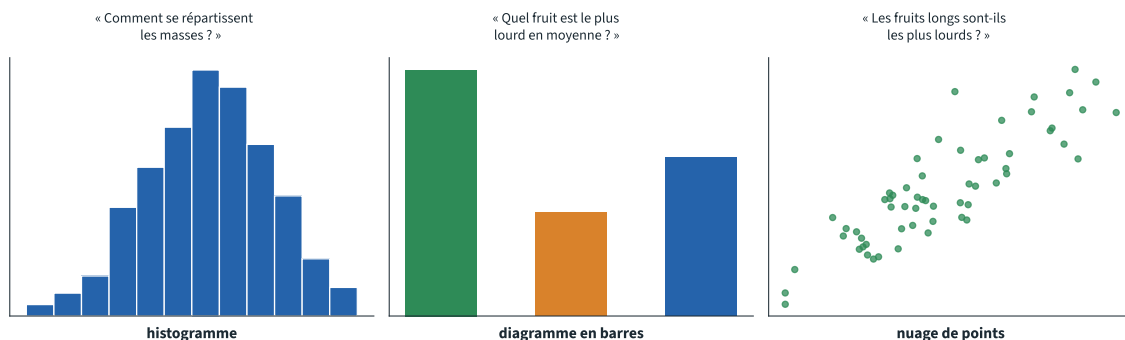
Un graphique sans titre ni unités est **illisible** : « 300 » de quoi ? des grammes, des francs, des fruits ?

- `title` : ce que l'on regarde.
- `xlabel` et `ylabel` : ce que représente chaque axe, **avec l'unité**.
- `legend` : à quoi correspond chaque couleur, dès qu'il y a des groupes.

Trois lignes de code, mais elles font la différence entre une image et une **information**.

Choisir le bon graphique

Le graphique dépend de la question posée



On ne choisit pas un graphique parce qu'il est joli, mais parce qu'il répond à la question que l'on se pose.

Trois questions, trois graphiques

- « **Comment se répartissent les masses ?** » — une seule mesure, on veut sa forme : **histogramme**.
- « **Quel fruit est le plus lourd en moyenne ?** » — une mesure comparée entre groupes : **diagramme en barres**.
- « **Les fruits longs sont-ils les plus lourds ?** » — deux mesures ensemble : **nuage de points**.

La bonne habitude

Formuler la question **en français d'abord**. Le choix du graphique en découle, et le code n'est plus qu'une formalité.

L'histogramme

Comprendre l'histogramme

L'idée en une phrase

L'histogramme découpe l'axe d'une mesure en **tranches** de même largeur, puis compte combien d'individus tombent dans chaque tranche. La hauteur d'une barre est donc un **effectif**.

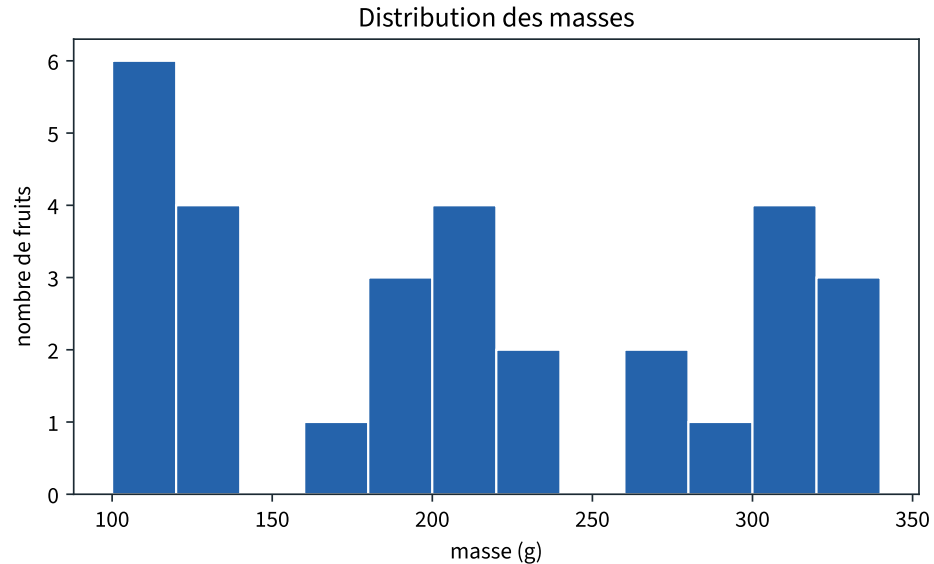
Une image

Comme si l'on rangeait les fruits dans des casiers étiquetés « 100–120 g », « 120–140 g »... puis qu'on comptait les fruits de chaque casier.

Attention : les barres se touchent, car la mesure est **continue** — c'est ce qui le distingue du diagramme en barres.

L'histogramme en pratique

```
plt.hist(df["masse"], bins=range(100, 360, 20))  
plt.xlabel("masse (g)")  
plt.ylabel("nombre de fruits")  
plt.title("Distribution des masses")
```



Lire cet histogramme

- L'axe des x porte la **masse** ; l'axe des y, le **nombre de fruits**.
- On distingue **trois amas** : vers 120 g, vers 200 g et vers 300 g.
- Entre ces amas, presque **aucun fruit** : les groupes sont bien séparés.

L'interprétation

Trois amas = trois familles de fruits. Ce sont les **bananes**, les **oranges** et les **mangues** — alors que la colonne **fruit** n'a jamais été utilisée pour tracer ce graphique.

Le rôle des tranches (**bins**)

- Des tranches **trop larges** lissent tout : les trois amas disparaissent en une seule barre.
- Des tranches **trop fines** font apparaître du bruit : chaque fruit devient sa propre barre.
- On ajuste jusqu'à voir la **structure** sans le bruit — ici, des tranches de 20 g.

C'est un **réglage**, pas une vérité : changer les tranches change ce que l'on voit.
Raison de plus pour toujours indiquer ses choix.

Le diagramme en barres

Comprendre le diagramme en barres

L'idée en une phrase

Le diagramme en barres compare **une valeur entre des groupes** : une barre par groupe, dont la hauteur est la valeur. Il part toujours d'une **agrégation** — une moyenne, une somme, un comptage.

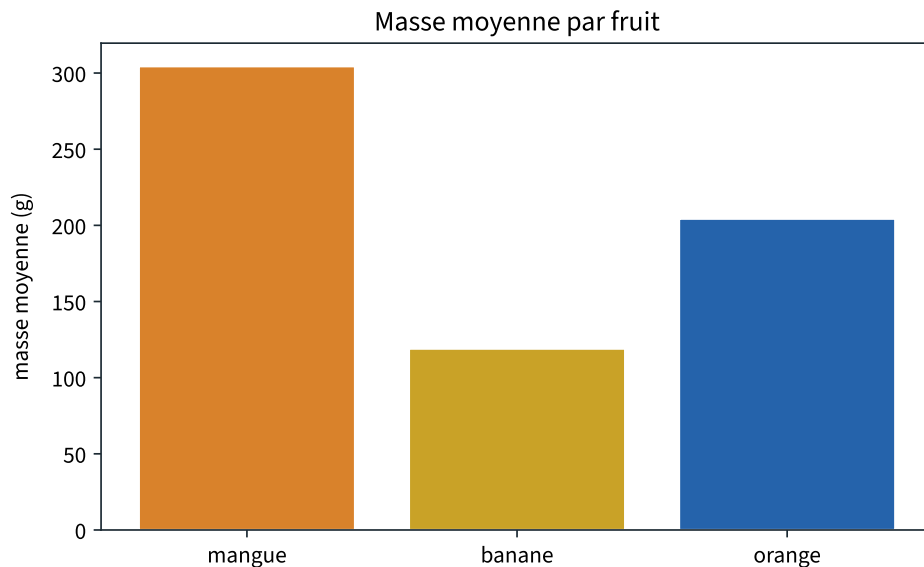
La démarche est en deux temps, déjà vus en Séance 7 :

Regrouper puis résumer : `groupby("fruit")["masse"].mean()`.

Tracer le résultat : une barre par fruit.

Le diagramme en barres en pratique

```
masse_moy = df.groupby("fruit")["masse"].mean()  
plt.bar(masse_moy.index, masse_moy.values)  
plt.ylabel("masse moyenne (g)")  
plt.title("Masse moyenne par fruit")
```



Lire ce diagramme

Les chiffres

Mangue : **304 g** en moyenne. Orange : **204 g**. Banane : **119 g**.

- La comparaison est **immédiate** : une mangue pèse environ deux fois et demie une banane.
- Mais attention : ce graphique montre des **moyennes**. Il ne dit rien de la dispersion à l'intérieur de chaque groupe.

C'est le complément de l'histogramme, pas son remplaçant : l'un montre la **forme**, l'autre compare des **résumés**.

Le nuage de points

Comprendre le nuage de points

L'idée en une phrase

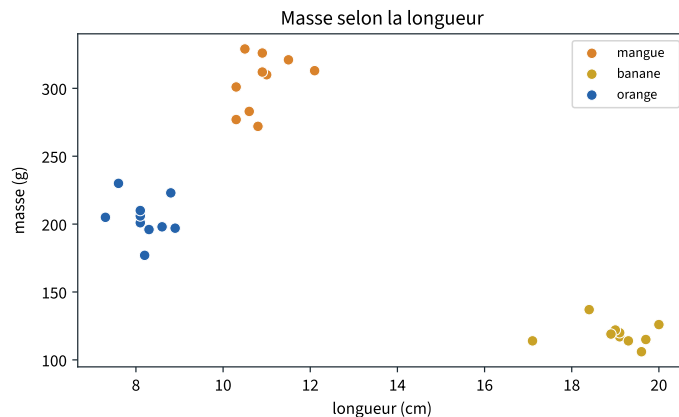
Le nuage de points place chaque individu selon **deux** mesures : une en abscisse, une en ordonnée. Chaque point est un fruit ; sa position résume ses deux mesures.

On y cherche deux choses :

- un **lien** : quand la longueur augmente, la masse augmente-t-elle aussi ?
- des **groupes** : les points forment-ils des paquets séparés ?

En **colorant par catégorie**, on répond aux deux questions d'un seul regard.

Le nuage révèle les fruits



Chaque point est un fruit, placé selon sa longueur et sa masse, coloré selon son type.

Trois **amas nettement séparés**, sans le moindre chevauchement.

Lire ce nuage, groupe par groupe

- La **banane** : longue (17–20 cm) mais légère (≈ 120 g) — en bas à droite.
- La **mangue** : courte (10–12 cm) et lourde (≈ 300 g) — en haut au milieu.
- L'**orange** : la plus courte (7–9 cm), de masse moyenne (≈ 200 g) — à gauche.

Le point décisif

Les trois groupes ne se touchent pas. Autrement dit : **connaître la longueur et la masse suffit à identifier le fruit** — sans le voir. C'est exactement ce que le Machine Learning va automatiser.

Une mini-EDA

La démarche d'analyse exploratoire

Quatre gestes, toujours les mêmes

L'**analyse exploratoire** (EDA) n'est pas une technique, c'est une **méthode** : comprendre les données avant de modéliser.

Charger : lire le fichier, vérifier `shape` — combien de lignes, de colonnes ?

Décrire : `describe`, `value_counts` — quelles valeurs, quels effectifs, des manques ?

Tracer : histogramme, barres, nuage — quelle forme, quels groupes ?

Observer : formuler en français ce que l'on a compris.

Le profil de chaque fruit

```
print(df.groupby("fruit")[["masse", "longueur", "sucre"]].mean().round(1))
```

RÉSULTAT →

	masse	longueur	sucre
fruit			
banane	119.0	19.0	11.8
mangue	304.4	10.9	14.0
orange	204.3	8.2	9.0

Trois lignes, trois colonnes : le **portrait chiffré** de chaque fruit.

L'étape « observer » : que dit ce tableau ?

- La **mangue** est la plus lourde (304 g) et la plus sucrée (14 %).
- La **banane** est de loin la plus longue (19 cm) et la plus légère.
- L'**orange** est la moins sucrée (9 %) et la plus courte.

Le point clé

Chaque mesure sépare les fruits, à sa manière. La masse distingue bien la mangue ; la longueur, la banane ; le sucre, l'orange. Un modèle exploitera les trois **ensemble**.

Le pont vers le ML

L'idée du Machine Learning, en une image

Ce que l'on vient de faire à la main

En regardant le nuage de points, **vous** avez appris à reconnaître un fruit : « long et léger → banane ». Le Machine Learning, c'est confier cet apprentissage à un programme.

Pour cela, on lui donne les données sous une forme précise :

- **x** — ce que l'on **observe** : les mesures (masse, longueur, sucre).
- **y** — ce que l'on veut **prédire** : le nom du fruit.

De la table aux objets X et y

X (les mesures)			y (a predire)
masse	longueur	sucres	fruit
301	10.3	14.7	mangue
137	18.4	11.1	banane
201	8.1	9.2	orange

Le modele apprend le lien $X \rightarrow y$: reconnaitre le fruit a partir de ses mesures.

Les colonnes de mesures forment X ; la colonne à prédire forme y .

Une ligne de X et la valeur de y en face décrivent **le même fruit**.

Construire X et y

```
X = df[["masse", "longueur", "sucre"]].values
y = df["fruit"].values
print("X shape :", X.shape)
print("y shape :", y.shape)
```

RÉSULTAT →

```
X shape : (30, 3)
y shape : (30,)
```

X est une **matrice** (30, 3) : 30 fruits en lignes, 3 mesures en colonnes. y est un **vecteur** (30,) : un nom par fruit. Les deux ont forcément **le même nombre de lignes**.

La suite, dans le cours de Machine Learning

Sous cette forme, `X` et `y` sont exactement ce qu'attend scikit-learn. Ce code n'est pas exécuté ici — il montre où l'on va :

```
modele = DecisionTreeClassifier()  
modele.fit(X, y)           # il APPREND le lien X -> y  
modele.predict([[300, 11, 14]]) # 300 g, 11 cm, 14 % : mangue ?
```

Deux mots pour finir

`fit` : le modèle **apprend** sur des fruits déjà étiquetés. `predict` : il **décide** pour un fruit nouveau. Tout le cours de Machine Learning consiste à comprendre ce qui se passe entre les deux.

Synthèse

Pièges fréquents

- **Conclure sur une moyenne** sans avoir regardé la répartition (souvenez-vous des deux paniers).
- **Oublier les légendes** : un graphique sans titre ni unités n'informe personne.
- **Choisir le graphique avant la question** : c'est la question qui commande.
- **Confondre les rôles** : x contient ce que l'on mesure, y ce que l'on cherche à prédire.

À retenir — Séance 8

L'essentiel

Pourquoi : une moyenne cache la forme des données ; un graphique la révèle.

Lequel : une mesure → histogramme ; une mesure par groupe → barres ; deux mesures → nuage de points.

Comment lire : titre, axes légendés avec unités, légende des couleurs ; puis chercher formes, groupes, liens.

Mini-EDA : charger → décrire → tracer → observer.

Pont ML : X (ce que l'on mesure) et y (ce que l'on prédit) ; **fit** pour apprendre, **predict** pour décider.

Bilan du cours

Le chemin parcouru

Des valeurs et des types (S1) aux structures (S2), au contrôle du flux (S3), aux fonctions (S4), aux objets (S5) ; puis à NumPy (S6), pandas (S7) et à la visualisation (S8).

Vous savez désormais **charger, explorer, résumer et représenter** des données — et vous comprenez ce qu'un modèle attend. Le cours de **Machine Learning** peut commencer.

Références

- Hunter, J. D. — *Matplotlib: A 2D Graphics Environment*, CiSE, 2007.
- VanderPlas, J. — *Python Data Science Handbook*, 2e éd., O'Reilly, 2023.
- McKinney, W. — *Python for Data Analysis*, 3e éd., O'Reilly, 2022.
- Tufte, E. — *The Visual Display of Quantitative Information*, 2e éd., 2001.

Ressources de la séance

- **Notebook** · [TP 8 — Visualiser et faire le pont](#) (116 Ko)