

CHAPITRE 2

MAITRISER LES MESURES DE DISPERSION ET DE POSITION

Ce que vous allez apprendre dans ce chapitre :

- Réviser les mesures de dispersion et de position
- Analyser les outliers
- Appliquer pratiquement avec Python/Excel



10 heures



CHAPITRE 2

MAITRISER LES MESURES DE DISPERSION ET DE POSITION

1. Rappeler les notions essentielles
- 2. Maitriser les mesures de dispersion et de position**
3. Assimiler les probabilités et distributions avancées
4. Vulgariser le langage R



01 – Maîtriser les mesures de dispersion et de position

Rappel des notions



Introduction

Les mesures centrales révèlent la tendance centrale ou la valeur centrale d'un ensemble de données.

Les mesures de dispersion décrivent la répartition ou la dispersion des valeurs dans un ensemble de données.

Les mesures de position indiquent la position relative d'une valeur par rapport aux autres dans un ensemble de données.

Mesures Centrales :
moyenne, médiane, mode

Mesures de Dispersion :
Étendue, Variance, Écart-type

Mesures de Position :
Quartiles, Déciles, Centiles

01 – Maîtriser les mesures de dispersion et de position

Rappel des notions



Étendue (Range) : Mesure de dispersion

L'étendue ou range en anglais, est une mesure de dispersion qui représente la différence entre la valeur maximale et la valeur minimale d'un ensemble de données.

$$E = x(\max) - x(\min)$$

$$E(\text{Suite Ord.}) = x(n) - x(1)$$

- Dans une suite ordonnée croissante, $x(1)$ et $x(n)$ sont respectivement l'observation extrême inférieure et supérieure

Interprétation :

Une étendue plus grande indique une dispersion plus importante des données, tandis qu'une étendue plus petite suggère une concentration autour de la moyenne

Limitations :

- Ne tient pas compte de la distribution interne des données.
- Peut être influencée par des valeurs aberrantes.

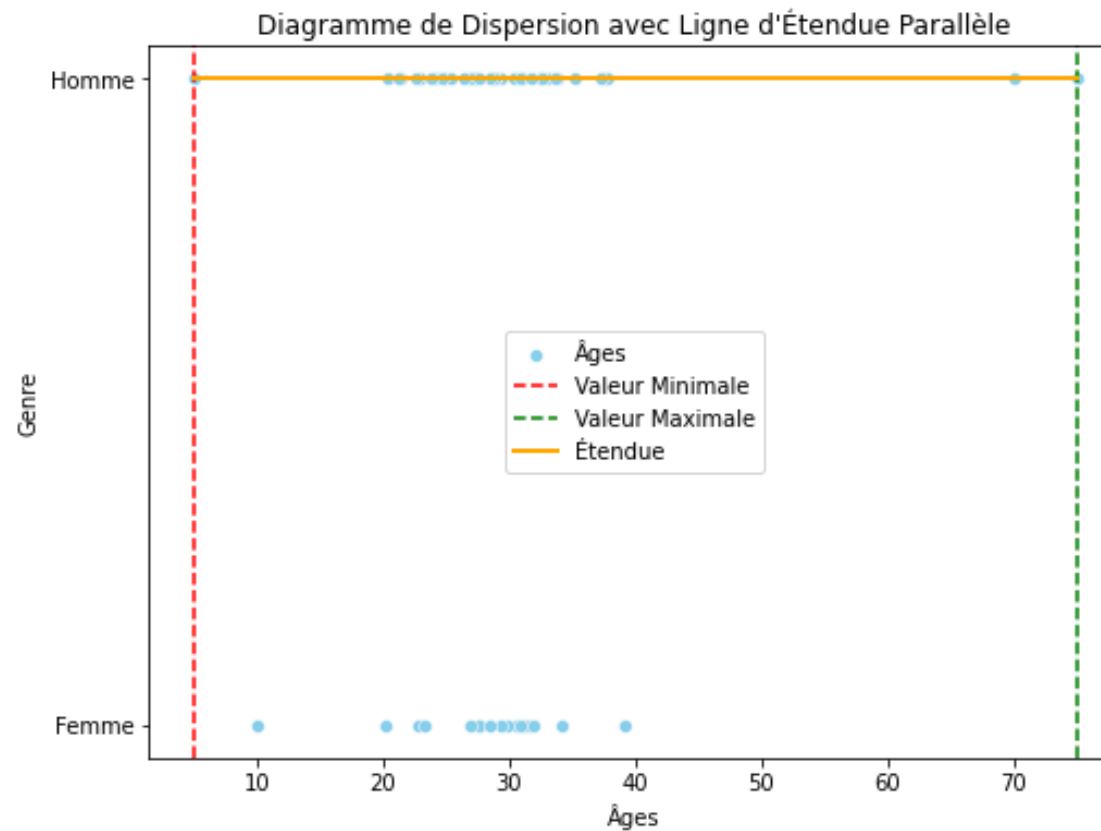
01 – Maîtriser les mesures de dispersion et de position

Rappel des notions



Étendue (Range) : Mesure de dispersion

Ce graphique de type scatter plot montre le genre en fonction de l'âge, avec une étendue de 70.00 unités, allant de 5.00 à 75.00.



La création et la configuration des graphiques seront expliquées dans les prochaines sections du cours.

01 – Maîtriser les mesures de dispersion et de position

Rappel des notions



Variance (Variation) : Mesure de dispersion

La variance σ^2 mesure la dispersion des valeurs au sein d'un ensemble de données. Elle est calculée en déterminant la moyenne des carrés des écarts **entre chaque valeur et la moyenne totale**. Pour calculer la variance de l'échantillon, divisez la somme des carrés des déviations par le nombre total de valeurs moins un

$$\sigma^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}$$

Interprétation :

Une variance élevée indique une dispersion importante, reflétant des valeurs plus éloignées de la moyenne, tandis qu'une faible variance suggère une concentration plus étroite autour de la moyenne.

Avantages :

Tient compte de la distribution interne des données.

Utile pour évaluer la variabilité.

Limitations :

Sensible aux valeurs aberrantes.

01 – Maîtriser les mesures de dispersion et de position

Rappel des notions



Variance (Variation) : Mesure de dispersion

Méthode de Calcul :

Pour calculer la variance pour les données (6, 7, 8), nous utiliserons la formule de la variance pour un échantillon puisque les données représentent un échantillon :

✓ Calcul de la Moyenne (\bar{X}) :

$$\bar{X} = \frac{\sum \text{Valeurs}}{\text{Nombre de Valeurs}}$$

$$\bar{X} = \frac{6 + 7 + 8}{3} = \frac{21}{3} = 7$$

✓ Calcul des Carrés des Déviations par rapport à la Moyenne de l'échantillon : $(X_i - \bar{X})^2$

$$\text{Pour 6 : } (6 - 7)^2 = 1$$

$$\text{Pour 7 : } (7 - 7)^2 = 0$$

$$\text{Pour 8 : } (8 - 7)^2 = 1$$

✓ Calcul de la Variance de l'échantillon :

$$\sigma^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1} \quad \longrightarrow \quad \sigma^2 = \frac{1 + 0 + 1}{3 - 1} = \frac{2}{2} = 1$$

Par conséquent, la variance de l'échantillon pour les données (6, 7, 8) est 1.

01 – Maîtriser les mesures de dispersion et de position

Rappel des notions

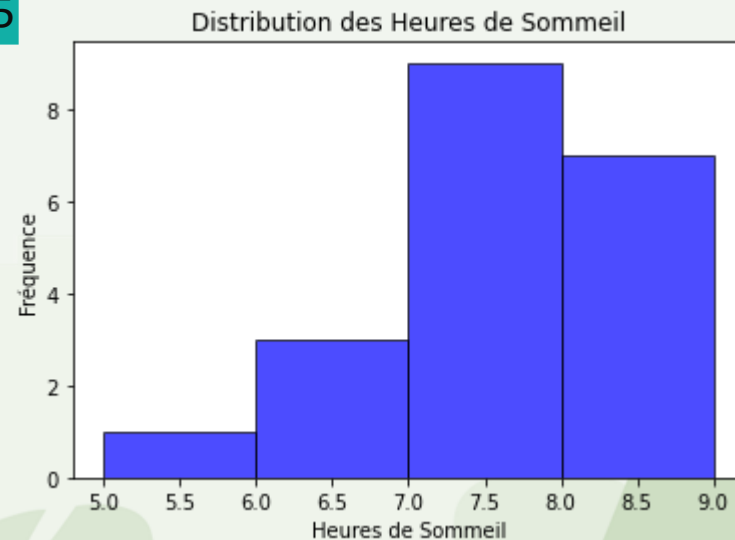


Variance (Variation) : Mesure de dispersion

Histogramme 1 - Distribution des Heures de Sommeil:

Moyenne: 7.25 Heures

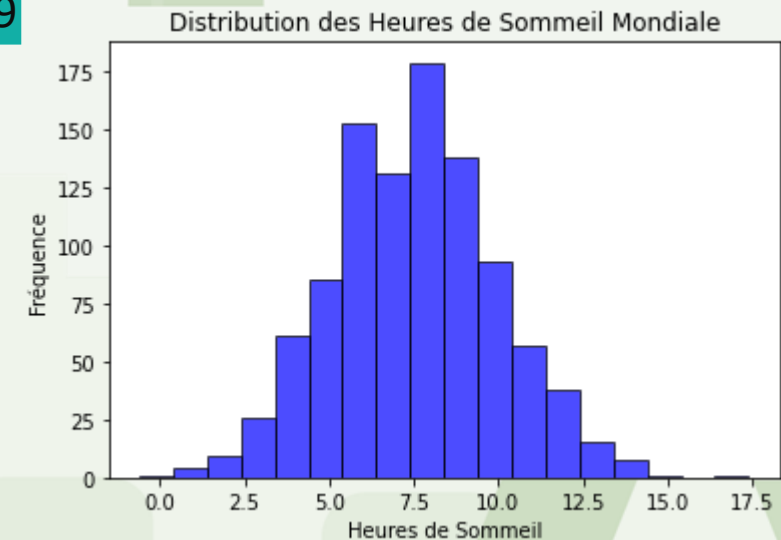
Variance: 1.0875



Histogramme 2 - Distribution des Heures de Sommeil Mondiale:

Moyenne: 7.55 Heures

Variance: 5.99



Interprétation :

Le premier histogramme illustre une **distribution localisée** avec une **faible variance** (1.0875), indiquant des habitudes de sommeil cohérentes autour de la moyenne. En revanche, le deuxième histogramme, représentant les heures de sommeil à l'échelle mondiale, présente une **distribution plus large** et plus diversifiée avec une **variance plus élevée** (5.99), soulignant une variabilité significative dans les habitudes de sommeil à l'échelle mondiale malgré une moyenne similaire.

La création de l'histogramme est expliquée au chapitre 1 : CRÉER DIVERS TYPES DE GRAPHIQUES du Deuxième partie du cours.

01 – Maîtriser les mesures de dispersion et de position

Rappel des notions



Ecart-type (standard deviation) (σ) : Mesure de dispersion

L'Écart-type ou standard deviation en anglais mesure la dispersion des valeurs au sein d'un ensemble de données. Il est calculé en déterminant la racine carrée de la variance, qui elle-même est obtenue en moyennant les carrés des écarts entre chaque valeur et la moyenne totale.

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}}$$

Interprétation :

Une valeur d'écart-type élevée indique une dispersion importante, signifiant que les valeurs sont plus éloignées de la moyenne. À l'inverse, un écart-type faible suggère une concentration plus étroite autour de la moyenne.

Avantages :

Mesure la dispersion des données dans les **mêmes unités que les valeurs d'origine**, ce qui facilite la compréhension de la variabilité des données.
Tient compte de la distribution interne des données.

Limitations :

Sensible aux valeurs aberrantes.

01 – Maitriser les mesures de dispersion et de position

Rappel des notions



Variance (σ^2) Vs Ecart-type (" σ ")

Caractéristique	Variance	Écart-type
Définition	Mesure l'écart moyen au carré par rapport à la moyenne.	La racine carrée de la variance, représente la distance moyenne entre chaque point de données et la moyenne.
Nature	En unités carrées des données d'origine.	Dans les mêmes unités que les données d'origine.
Unités	Carrées	Originales
Magnitude	Mesure la dispersion dans des unités carrées.	Fournit une mesure interprétable dans les unités d'origine.
Sensibilité	Moins sensible aux valeurs aberrantes.	Plus sensible aux valeurs aberrantes en raison de la racine carrée.

01 – Maîtriser les mesures de dispersion et de position

Rappel des notions



Quartiles: Mesure de position

Les quartiles sont des valeurs qui divisent un ensemble de données **triées** en quatre parties égales, chacune représentant 25% des données.

Calcul : Q_1 (premier quartile) est à 25%, Q_2 (deuxième quartile ou médiane) est à 50%, Q_3 (troisième quartile) est à 75%.

Interprétation : Q_1 représente le point où 25% des données sont inférieures, Q_2 **est la médiane**, et Q_3 représente le point où 75% des données sont inférieures.

Q1 : 25%

Premier quartile

Q2 : 50%

Médiane

Q3 : 75%

Troisième quartile

01 – Maîtriser les mesures de dispersion et de position

Rappel des notions



Déciles : Mesure de position

Les déciles sont des valeurs qui divisent un ensemble de données **triées** en dix parties égales, chaque décile représentant 10% des données.

Calcul : D_1 à D_9 représentent respectivement 10% à 90%, et D_{10} est équivalent au centile.

Interprétation : Les déciles permettent de visualiser la distribution des données en segments de 10%.

D1 : 10%

Premier décile

D2 : 20%

Deuxième décile

...

01 – Maitriser les mesures de dispersion et de position

Rappel des notions



Centiles : Mesure de position

Les centiles sont des valeurs qui divisent un ensemble de données **triées** en cent parties égales, chaque centile représentant 1% des données.

Calcul : C_1 à C_{99} représentent respectivement 1% à 99%, et C_{100} **est la valeur maximale.**

Interprétation : Les centiles permettent de quantifier la position d'une valeur spécifique par rapport à l'ensemble des données.

C1 : 1%
Premier centile

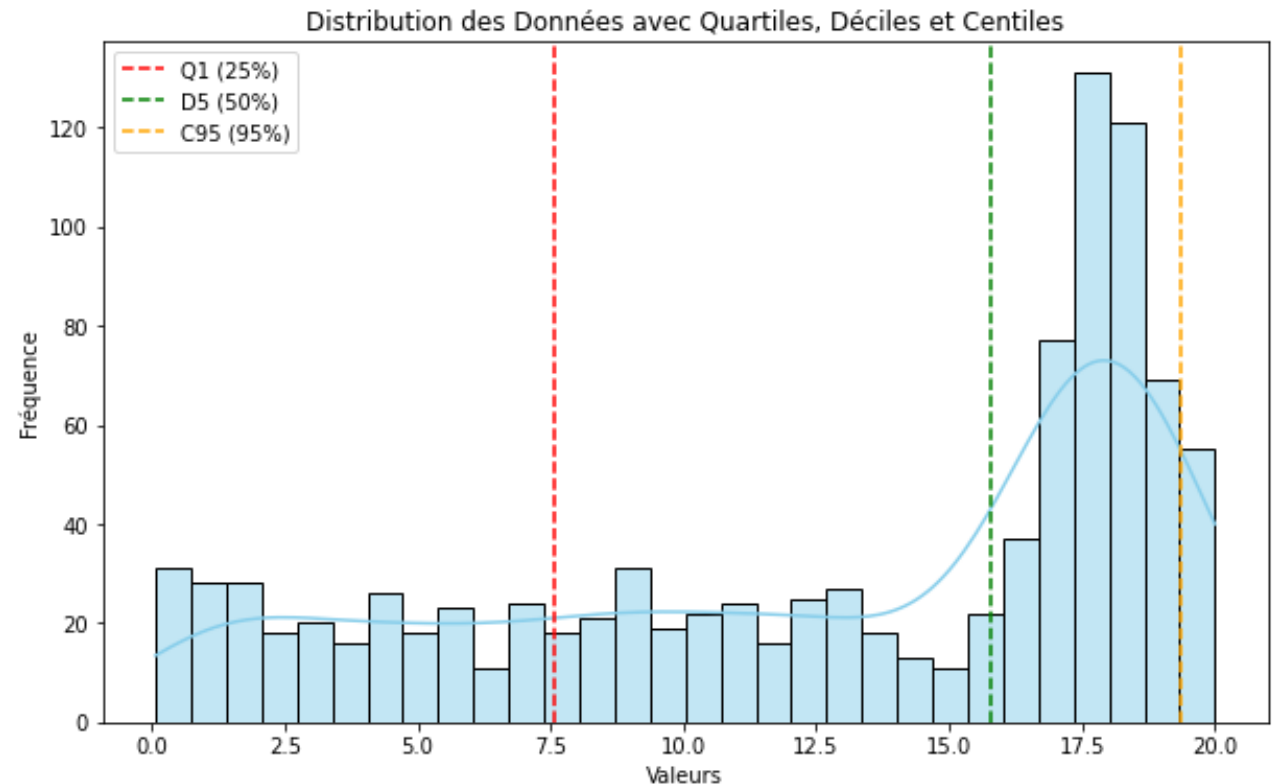
C2 : 2%
Deuxième centile

...

Exemple d'application aux résultats des étudiants : Interprétation et Analyse des Quartiles, Déciles et Centiles

Si Q_1 dans un ensemble de notes est égal à 7,5, cela signifie que 25% des étudiants ont obtenu 7,5 ou moins. Si D_5 est 16, alors 50% des données sont inférieures à 16. Enfin, si C_{95} est 19, cela indique que 95% des données sont inférieures à 19.

Ces termes permettent une compréhension détaillée de la distribution des données.



01 – Maîtriser les mesures de dispersion et de position

Rappel des notions



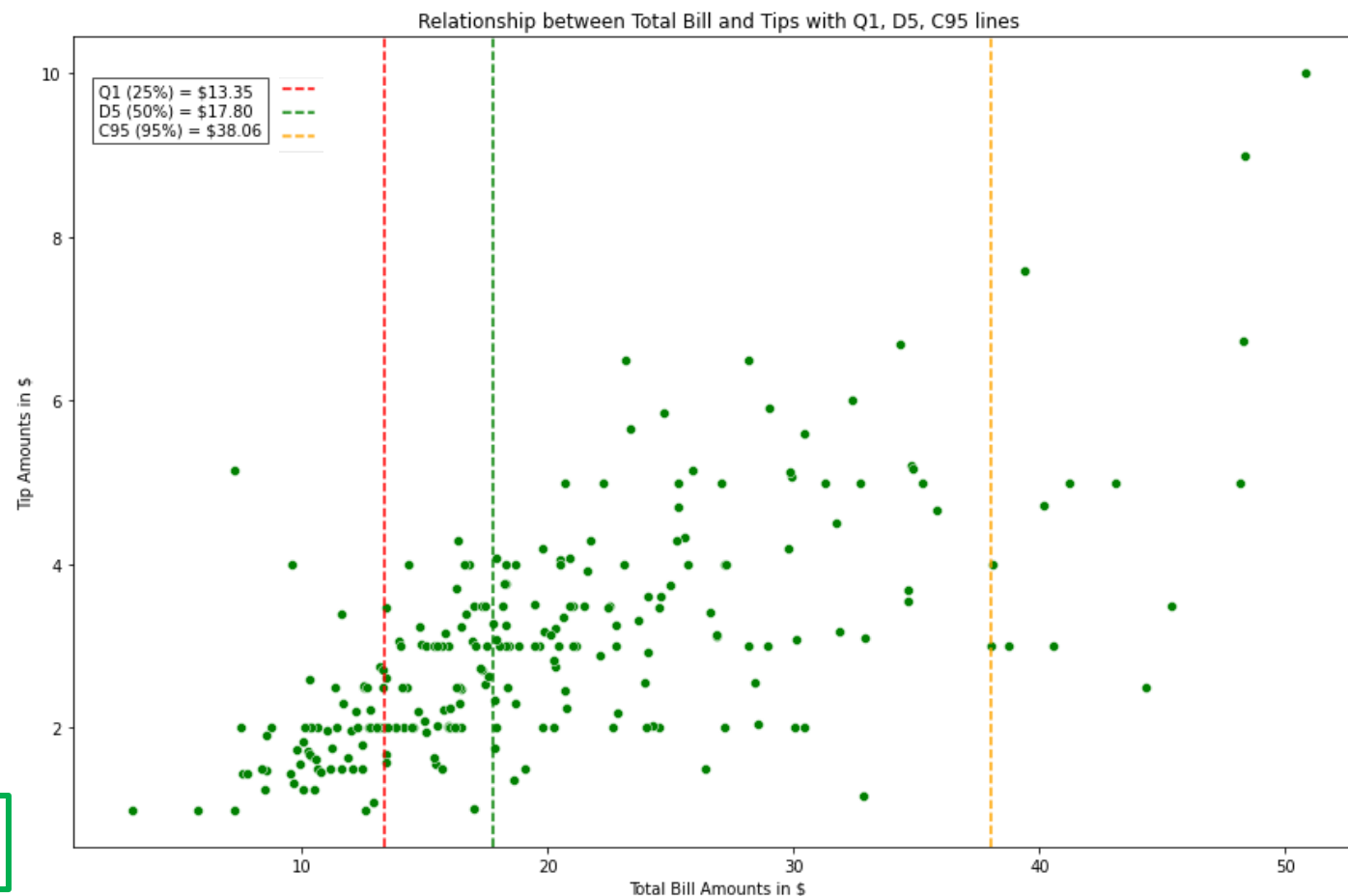
Exemple d'Application sur Tips Dataset : Visualisation des Quartiles, Déciles et Centiles

Pour aller plus loin dans notre exploration des statistiques, nous allons utiliser un ensemble de données réelles appelé 'Tips'. Ce jeu de données contient des informations sur les factures de restaurants, les pourboires et plus encore..

Cette approche, axée sur la compréhension des données, est appelée **l'analyse descriptive**.



La création de l'histogramme est expliquée au chapitre 1 : CRÉER DIVERS TYPES DE GRAPHIQUES du Deuxième partie du cours.



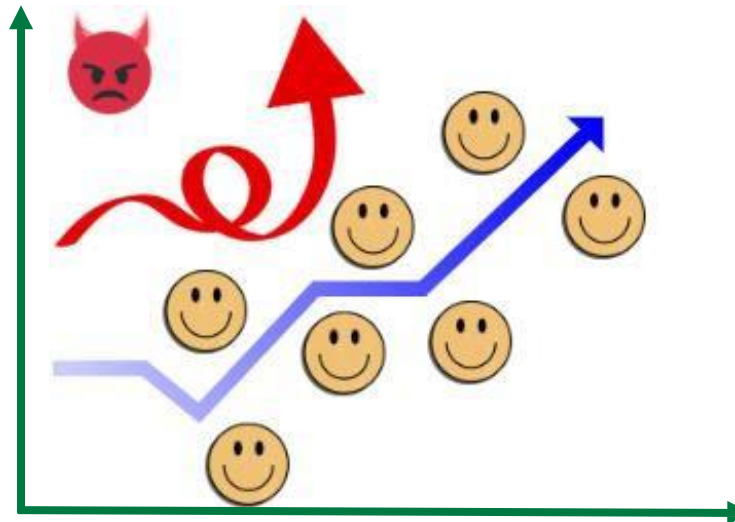
01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



Introduction :

En statistique Les valeurs aberrantes ou **outliers** en anglais, sont des points de données qui diffèrent significativement de la majorité des données dans un ensemble. Ce sont des observations qui se situent à une distance anormale par rapport aux autres valeurs, indiquant potentiellement un événement rare ou une erreur de mesure. Identifier et comprendre ces outliers revêt une importance cruciale dans l'analyse statistique, car ils peuvent exercer une influence sur les résultats et l'interprétation des analyses. Particulièrement dans des ensembles de données plus petits, leur influence peut être majeure.



In statistics, an outlier is a data point that differs significantly from other observations.

01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



Impact des valeurs aberrantes sur l'analyse et la modélisation des données :

Les valeurs aberrantes peuvent avoir un impact significatif sur les résultats de l'analyse et de la modélisation des données. Elles peuvent fausser les mesures statistiques, conduisant à des estimations incorrectes de la tendance centrale et de la dispersion.

Par exemple, si un ensemble de données sur les salaires des employés contient quelques salaires extrêmement élevés en raison de primes de direction, le salaire moyen peut être considérablement **surestimé**, entraînant une moyenne salariale exagérée pour l'organisation.

De même, si un ensemble de données sur les prix immobiliers contient quelques prix très bas en raison d'**erreurs de saisie**, le prix médian peut ne pas représenter avec précision le prix typique des maisons dans cette région.

Les outliers peuvent également affecter les **performances des modèles de ML**. Certains modèles, tels que la régression linéaire, sont sensibles aux outliers.



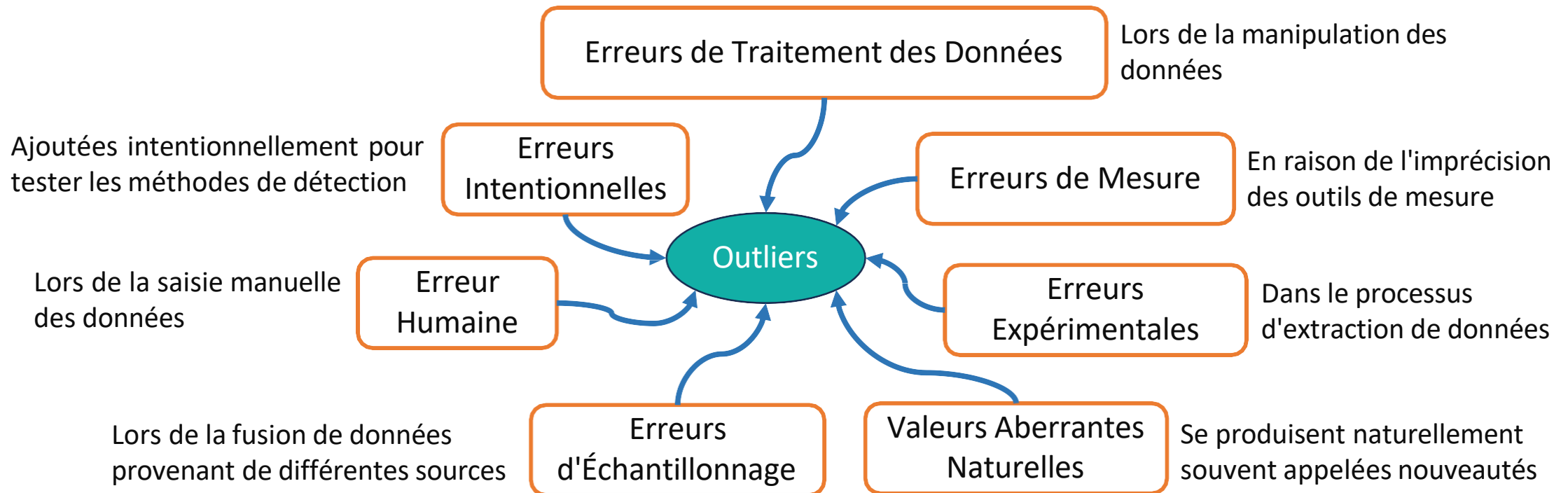
01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



Causes des outliers ou valeurs aberrantes

Comprendre les causes des valeurs aberrantes **aide grandement à les traiter**, à **prendre des décisions éclairées** et à **réduire leur occurrence** lors des observations à venir.



Analyse et identification des Outliers

L'identification des outliers est un processus crucial dans l'analyse de données pour repérer les valeurs aberrantes. Dans ce cours, nous proposons trois méthodes principales :

- **Mesures Statistiques** : Utilisation des mesures telles que le minimum et le maximum.
- **Visualisation** : Emploi de graphiques tels que les box plots et même les tableaux d'Excel pour une identification visuelle.
- **Techniques Statistiques Avancées** : Mise en œuvre de méthodes comme DBSCAN pour une identification précise.

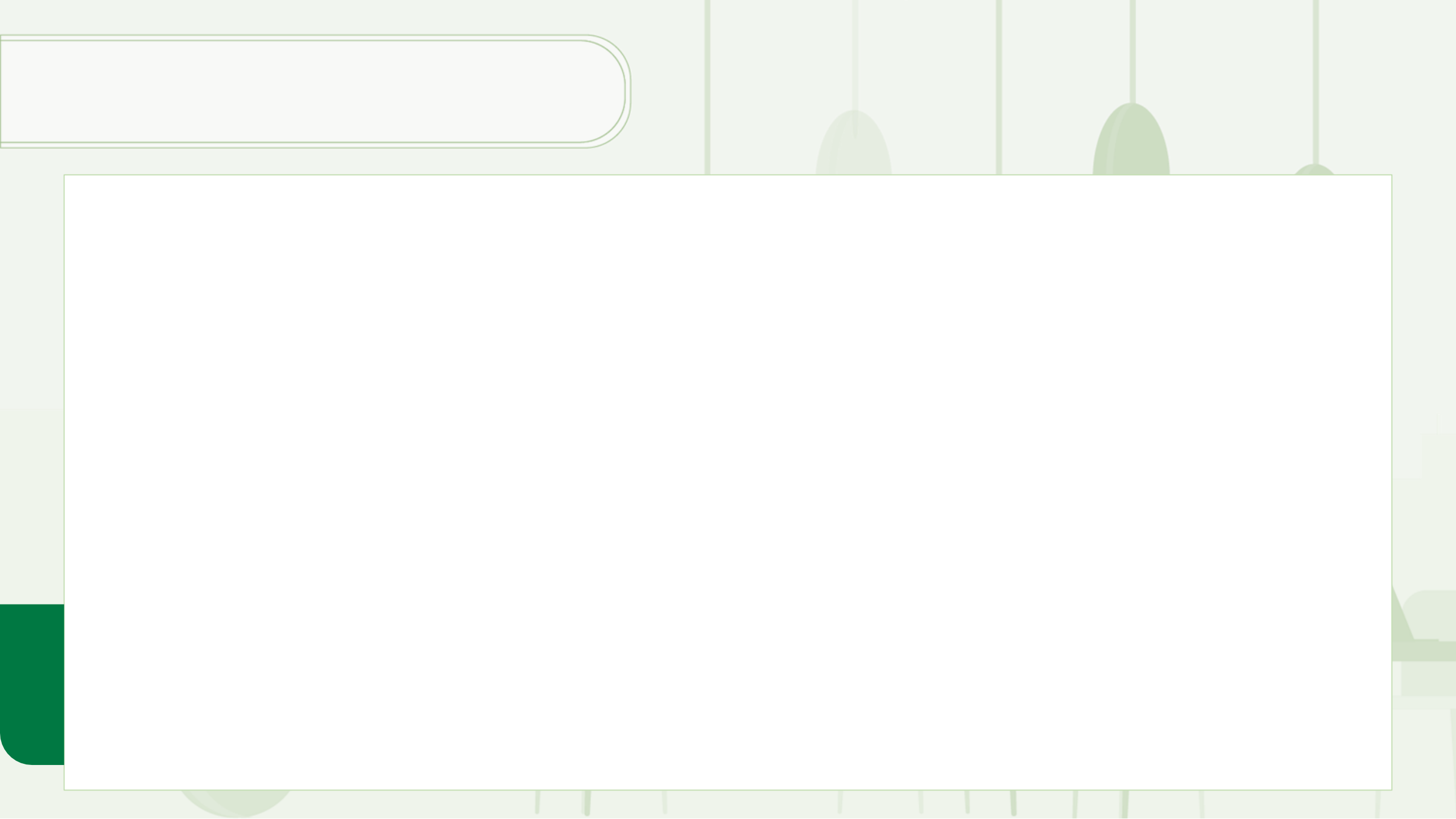


Gestion et traitement des outliers

Explorez le processus de gestion des valeurs aberrantes dans l'analyse de données. Trois éléments clés guideront cette gestion :

- **Stratégies de remplacement** : Explorez des stratégies telles que le remplacement par des valeurs moyennes ou médianes pour des analyses plus robustes.
- **Supprimer les outliers** : Déterminez judicieusement quand supprimer les valeurs aberrantes de l'ensemble de données.
- **Acceptez la présence d'outliers lorsque justifié** : Discernez les situations où la présence d'outliers est pertinente et peut apporter des insights significatifs.





01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



Mesures statistiques : Méthode des deux écarts-types (Two standard deviations method)

La méthode des deux écarts-types est une technique statistique qui identifie les valeurs aberrantes en considérant celles qui se situent en dehors de l'intervalle défini par **la moyenne plus ou moins deux fois l'écart-type**, dans le cadre d'une **distribution normale**. Elle repose sur l'idée que la plupart des données dans une distribution normale se trouvent dans l'intervalle de deux écarts-types de la moyenne, et donc, les valeurs en dehors de cet intervalle peuvent être considérées comme des valeurs aberrantes.

Le calcul se fait à l'aide de la fonction : $\text{Seuil} = \bar{X} (+ \text{ou} -) 2 \sigma$

Application de la méthode : Exemple des revenus mensuels

Génération des données :

Revenus mensuels : [4900, 4800, 5200, 5100, 4900, 4800, 5000, 15000, 4900, 4800, 1000, ...] (100 valeurs au total)

Calcul des mesures statistiques :

Moyenne :

$$\bar{X} = \frac{\sum \text{Valeurs}}{\text{Nombre de Valeurs}}$$

Écart-type :

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}}$$

$$\bar{X} = (4900 + 4800 + 5200 + \dots + 1000) / 100 \Rightarrow \bar{X} = 4966$$

$$\sigma = \sqrt{((4900 - 4966)^2 + (4800 - 4966)^2 + \dots + (1000 - 4966)^2) / 99} \Rightarrow \sigma = 1407$$

01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



Mesures statistiques : Méthode des deux écarts-types

Identification des seuils supérieur et inférieur :

$$\text{Seuil Supérieur : } Ss = \bar{X} + 2\sigma$$

$$\text{Seuil Supérieur} = 4966 + 2 \times 1407 = 7781$$

$$\text{Seuil Inférieur : } Si = \bar{X} - 2\sigma$$

$$\text{Seuil Inférieur} = 4966 - 2 \times 1407 = 2152$$

Identification des Outliers :

Les valeurs aberrantes « Outliers » sont celles qui sont au-dessus du seuil supérieur ou en dessous du seuil inférieur

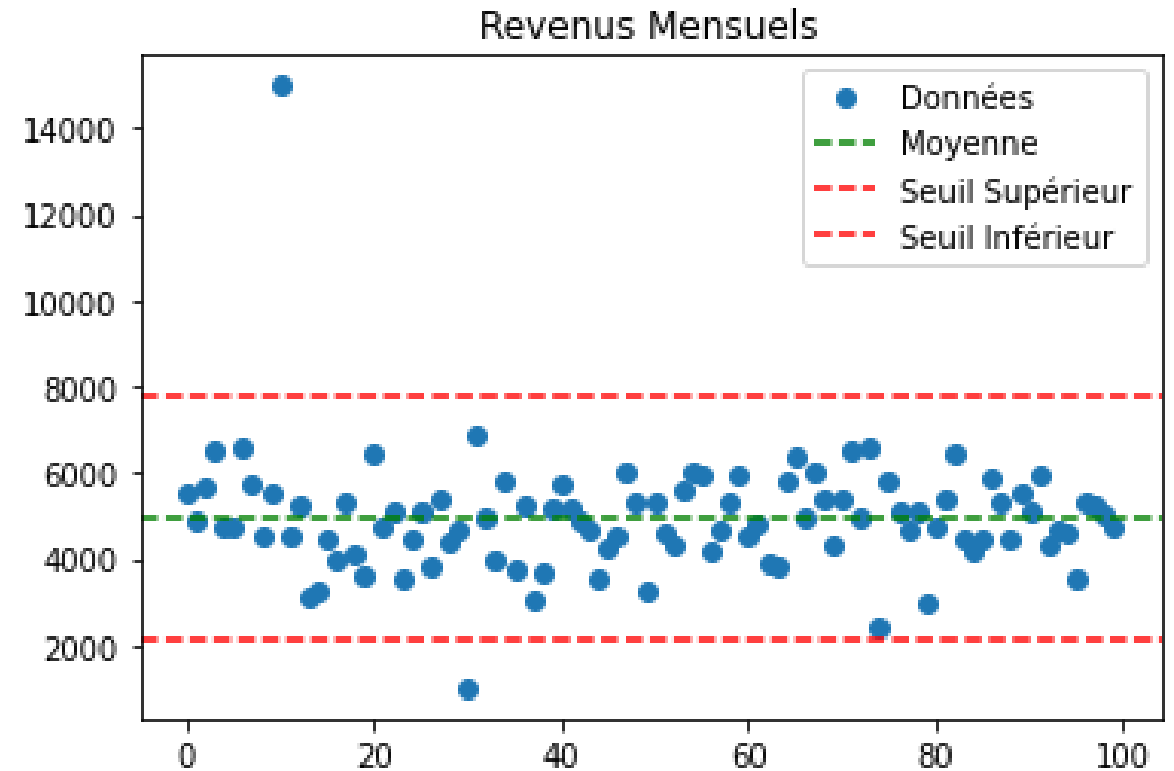
Affichage des résultats :

Revenus mensuels : [4900, 4800, 5200, 5100, 4900, 4800, 5000, 15000, 4900, 4800, 1000, ...]

Moyenne : 4966 | Écart-type : 1407

Seuil Supérieur : 4966 | Seuil Inférieur : 2152

Ex. Valeurs Aberrantes : [15000, 1000]



01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification des outliers : Exemple de calcul de la méthode des deux écarts-types avec Python

Génération des données :

Considérons un ensemble de données avec 10 valeurs :

```
import numpy as np
```

```
data = np.array([15, 22, 18, 25, 20, 17, 23, 21, 19, 30])
```

Calcul des mesures statistiques :

Étape 1 : Calcul de la Moyenne et de l'Écart-type

```
mean = np.mean(data)
```

```
std_dev = np.std(data)
```

Étape 2 : Définition des Seuils

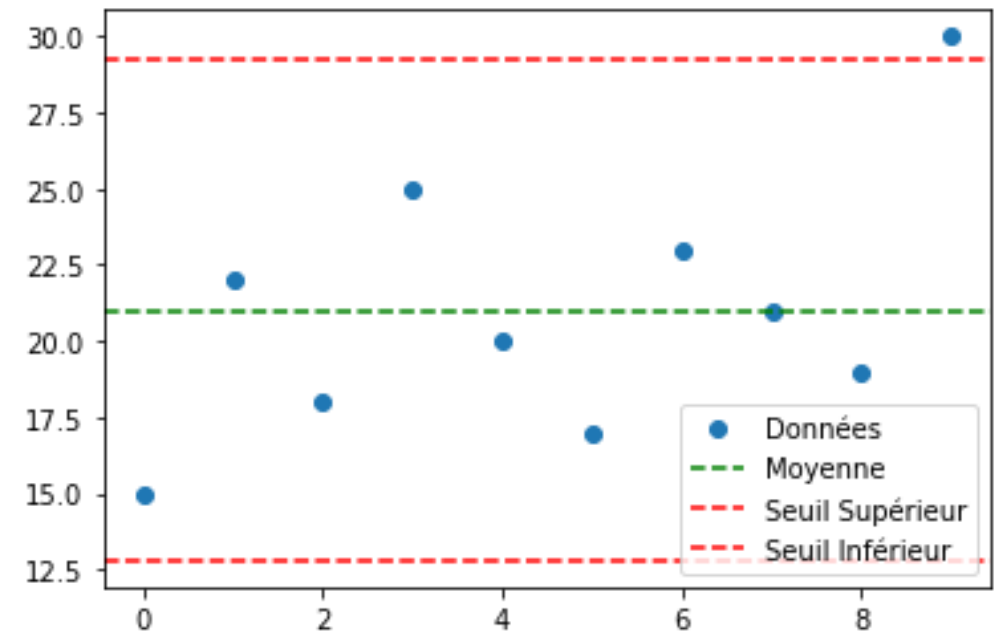
```
threshold_upper = mean + 2 * std_dev
```

```
threshold_lower = mean - 2 * std_dev
```

Étape 3 : Identification des Valeurs Aberrantes

```
outliers = [value for value in data if value > threshold_upper or value < threshold_lower]
```

Affichage des résultats :



```
Données : [15 22 18 25 20 17 23 21 19 30]
Moyenne : 21.0 Écart-type : 4.09878030638384
Seuil supérieur : 29.19756061276768
Seuil inférieur : 12.80243938723232
Valeurs aberrantes : [30]
```


01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification des outliers : Affichage de la méthode des deux écarts-types avec Python

Reference d'affichage numériques :

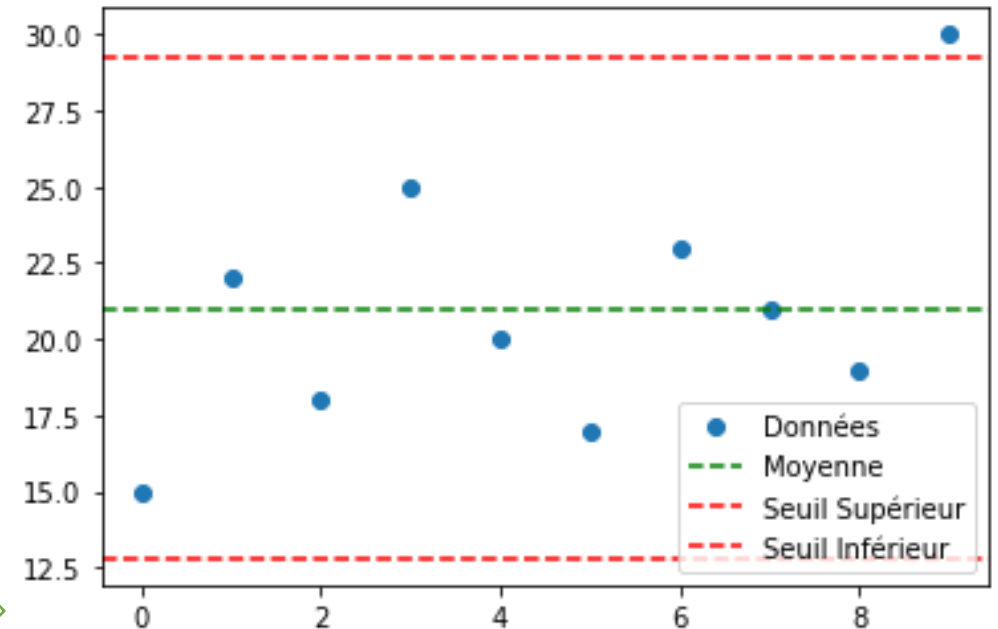
```
print("Données :", data)
print("Moyenne :", mean)
print("Écart-type :", std_dev)
print("Seuil supérieur :", threshold_upper)
print("Seuil inférieur :", threshold_lower)
print("Valeurs aberrantes :", outliers)
```

Reference d'affichage graphique :

```
import matplotlib.pyplot as plt
# Visualisation des données avec seuils
plt.plot(data, 'o', label='Données')
plt.axhline(y=mean, color='g', linestyle='--', label='Moyenne')
plt.axhline(y=threshold_upper, color='r', linestyle='--', label='Seuil Supérieur')
plt.axhline(y=threshold_lower, color='r', linestyle='--', label='Seuil Inférieur')
plt.legend()
plt.show()
```

Sortie Attendue :

```
Données : [15 22 18 25 20 17 23 21 19 30]
Moyenne : 21.0 Écart-type : 4.09878030638384
Seuil supérieur : 29.19756061276768
Seuil inférieur : 12.80243938723232
Valeurs aberrantes : [30]
```



01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



Mesures statistiques :

Méthode IQR (Interquartile Range) : $IQR = Q3 - Q1$

L'intervalle interquartile (IQR) est une mesure statistique qui évalue la dispersion des données au sein d'un ensemble. Il est défini comme la différence entre le troisième quartile (Q3) et le premier quartile (Q1). Utilisé pour détecter les valeurs aberrantes, l'IQR identifie les points de données situés en dehors d'une plage définie par 1,5 (K) fois la longueur de l'IQR au-dessus du troisième quartile et en dessous du premier quartile. Cela permet de cibler les observations inhabituelles dans un ensemble de données.

$$\text{Seuil inférieure} = Q1 - 1,5 \times IQR \quad \text{Seuil supérieure} = Q3 + 1,5 \times IQR$$

Méthode Tukey :

L'utilisation de $K = 3$ au lieu de 1,5 signifie l'application de la méthode de Tukey, qui a tendance à éliminer moins de données.

Application de la méthode IQR :

Les Etapes à suivre :

1. Calculez le premier quartile (Q1) et le troisième quartile (Q3) de vos données.
2. Calculez l'écart interquartile (IQR) en soustrayant Q1 de Q3 : $IQR = Q3 - Q1$.
3. Définissez une limite inférieure comme $Q1 - 1,5 \times IQR$ et une limite supérieure comme $Q3 + 1,5 \times IQR$.
4. Toute valeur en dehors de ces limites est considérée comme une valeur aberrante.



Les valeurs couramment utilisées pour k sont 1.5 et 3, bien que d'autres valeurs puissent également être utilisées en fonction des besoins.

01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification des outliers : Application de la méthode IQR et Tukey utilisant Python

Pour détecter les Outliers dans un ensemble de données, la méthode statistique IQR évalue la dispersion des données en identifiant les points situés en dehors d'une plage définie par le premier et le troisième quartile, offrant une approche efficace pour cibler les observations inhabituelles.

```
import numpy as np
import matplotlib.pyplot as plt

# Exemple de données
data = np.array([2, 3, 4, 5, 6, 7, 20])

# Calcul de Q1, Q3 et IQR
q1 = np.percentile(data, 25)
q3 = np.percentile(data, 75)
iqr = q3 - q1

# Calcul des limites pour la méthode IQR
iqr_limite_inf = q1 - 1.5 * iqr
iqr_limite_sup = q3 + 1.5 * iqr

# Calcul des limites pour la méthode de Tukey
tukey_limite_inf = q1 - 3 * iqr
tukey_limite_sup = q3 + 3 * iqr
```

```
# Identification des valeurs aberrantes
iqr_outliers = data[(data < iqr_limite_inf) | (data > iqr_limite_sup)]
tukey_outliers = data[(data < tukey_limite_inf) | (data > tukey_limite_sup)]
```

```
# Affichage des valeurs calculées
print(f"Q1 : {q1}, Q3 : {q3}, IQR : {iqr}")
print(f"Limite inférieure IQR : {iqr_limite_inf}, Limite supérieure IQR : {iqr_limite_sup}")
print(f"Limite inférieure Tukey : {tukey_limite_inf}, Limite supérieure Tukey : {tukey_limite_sup}")
print(f"Valeurs aberrantes IQR : {iqr_outliers}")
print(f"Valeurs aberrantes Tukey : {tukey_outliers}")
```

Sortie Attendue :

```
Q1 : 3.5, Q3 : 6.5, IQR : 3.0
Limite inférieure IQR : -1.0, Limite supérieure IQR : 11.0
Limite inférieure Tukey : -5.5, Limite supérieure Tukey : 15.5
Valeurs aberrantes IQR : [20]
Valeurs aberrantes Tukey : [20]
```

01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification des outliers : Visualisation des mesures IQR et Tukey utilisent Python

Le graphique de visualisation présente les données initiales avec des points de données en bleu, les quartiles Q1 et Q3 en vert et rouge respectivement, ainsi que les limites définies par la méthode IQR et la méthode de Tukey. Les valeurs aberrantes détectées par chaque méthode sont marquées en rouge (IQR) ou orange (Tukey). Cette visualisation permet une compréhension rapide des points atypiques dans le jeu de données.



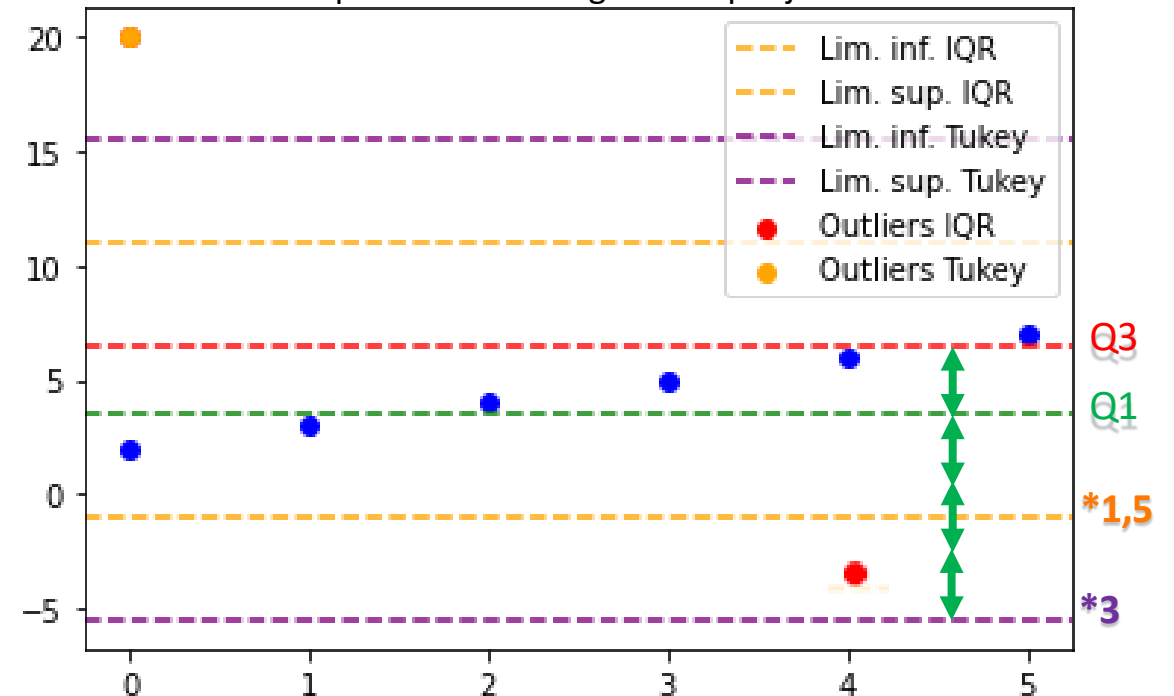
Des données bien nettoyées et propres sont essentielles pour renforcer la détection des intrusions et améliorer la fiabilité des systèmes.



La diapositive suivante présente les étapes à suivre pour générer ce graph de type scatter.

les types de graphiques et leur création sont abordés dans d'autres sections suivantes et intégrés en parallèle avec le cours.

Température du réfrigérateur par jour



01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification des outliers : Visualisation des mesures IQR et Tukey utilisant Python

Pour référence, nous présentons le code python nécessaire à la création de graphique précédente.

```
# Création d'un masque pour afficher uniquement les valeurs correctes
correct_data_mask = ~np.isin(data, iqr_outliers) & ~np.isin(data, tukey_outliers)

# Nuage de points avec uniquement les valeurs correctes
plt.scatter(range(len(data[correct_data_mask])), data[correct_data_mask], color='blue')#,
label='Points de données'
plt.axhline(y=q1, color='green', linestyle='--') #, label='Q1'
plt.axhline(y=q3, color='red', linestyle='--') #, label='Q3'
plt.axhline(y=iqr_limite_inf, color='orange', linestyle='--', label='Lim. inf. IQR')
plt.axhline(y=iqr_limite_sup, color='orange', linestyle='--', label='Lim. sup. IQR')
plt.axhline(y=tukey_limite_inf, color='purple', linestyle='--', label='Lim. inf.
Tukey') plt.axhline(y=tukey_limite_sup, color='purple', linestyle='--', label='Lim. sup.
Tukey') plt.scatter(range(len(iqr_outliers)), iqr_outliers, color='red', label='Outliers
IQR')
plt.scatter(range(len(tukey_outliers)), tukey_outliers, color='orange', label='Outliers
Tukey')

# Affichage du graphique
plt.legend()
plt.show()
```

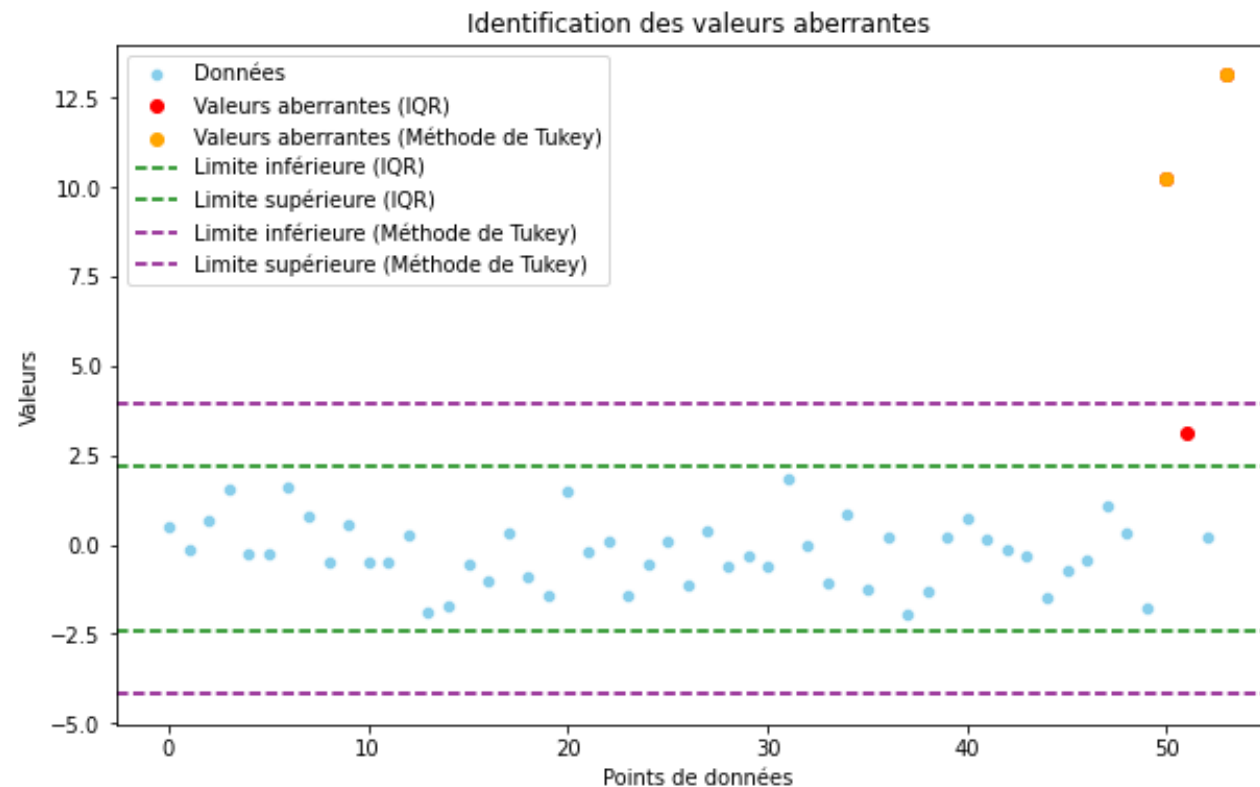
01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification des outliers : 2^{ème} Visualisation des mesures IQR et Tukey

Le graph de visualisation présente les limites définies par la méthode IQR et la méthode de Tukey. Les valeurs aberrantes détectées par chaque méthode sont marquées en rouge (IQR) et orange (Tukey). Cette visualisation simple est suffisante pour la détection des points atypiques dans le jeu de données.



Identification par visualisation: Box plot un outil efficace pour identifier les valeurs aberrantes

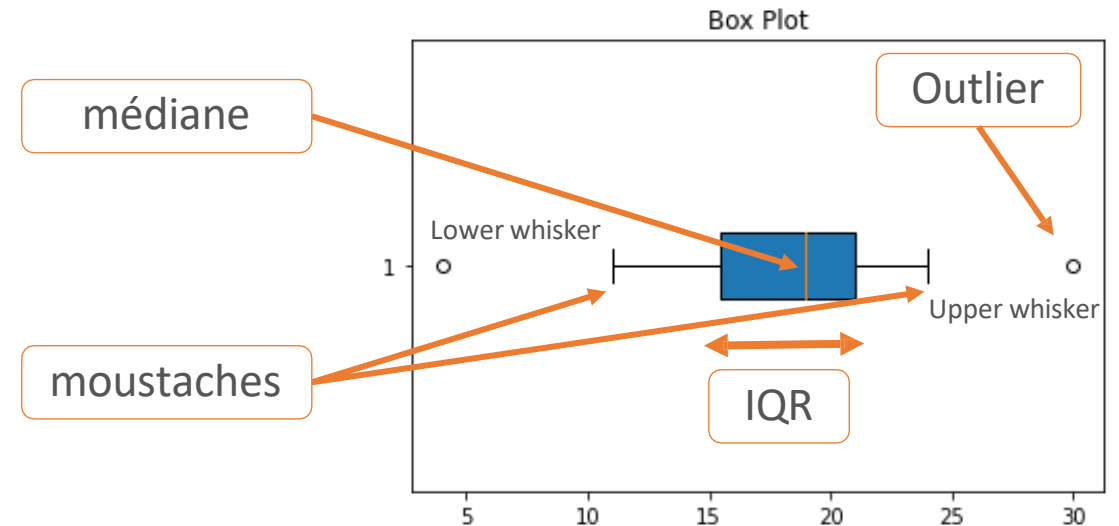
L'utilisation de graphiques, tels que les box plots et scatter plots, est une méthode puissante pour détecter visuellement les valeurs aberrantes. Les box plots présentent une représentation graphique des quartiles, de la médiane et des valeurs extrêmes, offrant une vue d'ensemble de la distribution des données.

Box Plots:

Les box plots permettent de repérer les valeurs aberrantes de manière intuitive. La boîte représente l'étendue interquartile IQR (**Interquartile Range**) entre le premier quartile (Q1) et le troisième quartile (Q3). Les lignes (moustaches) s'étendent jusqu'aux valeurs les plus éloignées qui ne sont pas considérées comme des valeurs aberrantes.

Interprétation des Box Plots :

Les box plots visualisent la distribution des données, montrant la médiane, les quartiles, et les valeurs minimale et maximale. Ils sont utiles pour comparer visuellement les ensembles de données. Les outliers sont représentés à l'extérieur des moustaches.



01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



Identification par visualisation : Interprétation des Box Plots

Lecture d'un Box Plot :

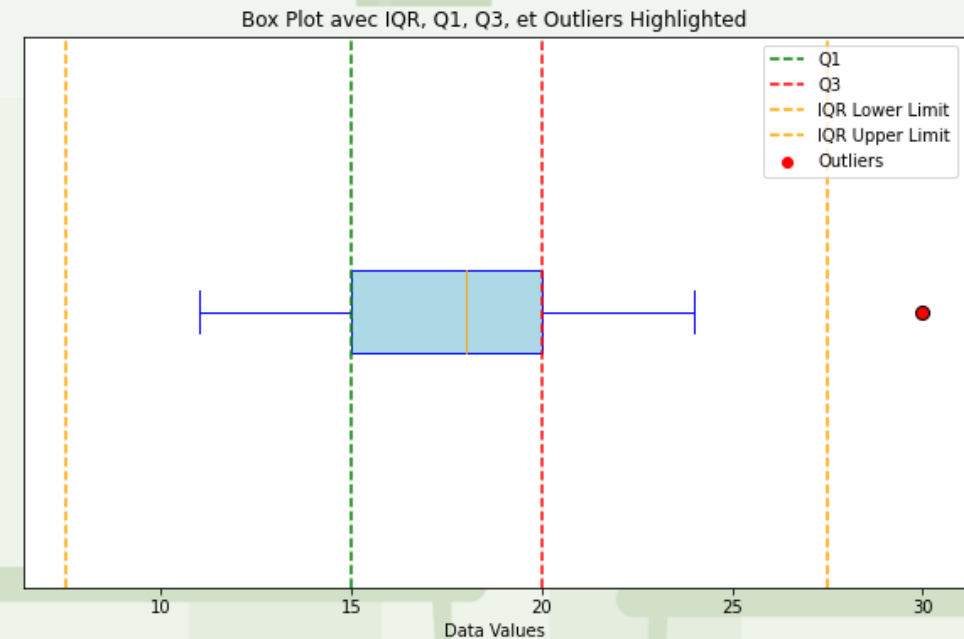
- Minimum et maximum alignés aux extrémités.
- Quartiles inférieur et supérieur alignés aux bords de la boîte.
- Médiane alignée avec la ligne intérieure de la boîte.

Analyse d'un Box Plot :

- **Médiane** plus grande indique une moyenne plus grande.
- **Étendue** = maximum – minimum. Une plus grande étendue signifie une dispersion plus importante.
- **IQR** = $Q3 - Q1$. Un IQR plus grand signifie une dispersion accrue pour la moitié centrale des données.

Outliers par box plot :

Valeurs aberrantes **à l'extérieur** des moustaches.



Rappel IQR method :

$iqr_lower_limit = q1 - 1.5 * iqr$

$iqr_upper_limit = q3 + 1.5 * iqr$

01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



Techniques statistiques avancées : Z-Score (Standardization)

Le **Z-score**, ou standard score, est une mesure statistique qui quantifie de **combien d'écart-types un point de données se trouve par rapport à la moyenne** d'un ensemble de données. Il est exprimé en tant que nombre d'écart-types par rapport à la moyenne, indiquant ainsi la position relative d'une observation ou d'un point de données particulier.

Formule Population : $Z = (x - \mu) / \sigma$

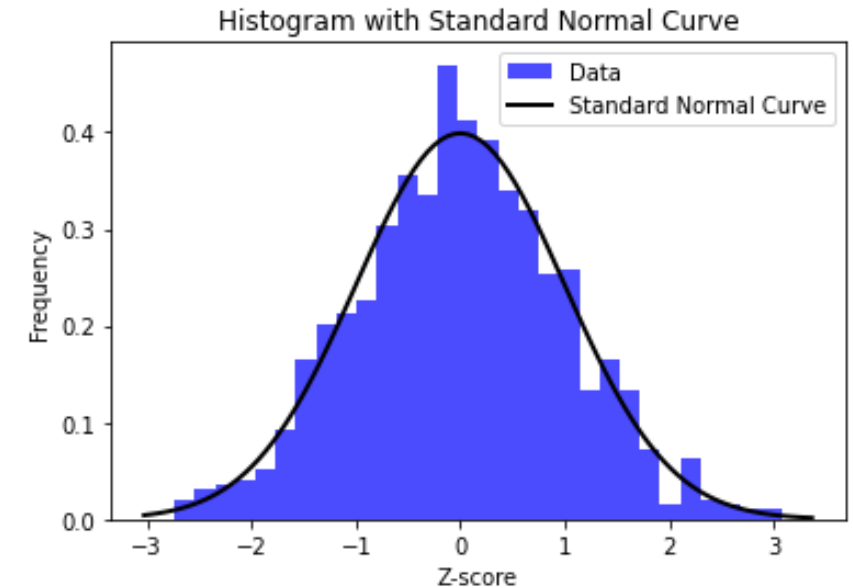
La formule pour calculer le Z-score d'un point de données 'x' dans un ensemble de données avec une moyenne ' μ ' et un écart-type ' σ '.

Formule Échantillon : $Z = (x - \bar{x}) / s$

utilisée lorsqu'on travaille avec un échantillon plutôt qu'une population. Où \bar{x} est la moyenne de l'échantillon et 's' l'écart-type de l'échantillon.

Interprétation :

Un Z-score de 0 signifie que le point de données est à la moyenne, les Z-scores **positifs** et **négatifs** indiquent une position **au-dessus** ou **en dessous** de la moyenne. Ils **identifient les valeurs aberrantes**, permettent la **comparaison entre distributions** et **définissent la position** relative des données.



01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



Techniques statistiques avancées : Z-Score Exemples de compréhension

Exemple de Calcul dans un Échantillon :

Supposons un échantillon de données $X=\{4,5,7,,\}$

Avec une moyenne d'échantillon $\bar{X}=5$

Un écart-type d'échantillon $s=1$.

Pour calculer le Z-score du point de données 7, utilisez la formule :

$$Z=(7 - 5) / 1 = 2$$

Les étapes restent les mêmes, mais la formule est ajustée pour refléter l'utilisation des statistiques d'échantillon.



Rappel :

Un échantillon est un ensemble d'individus représentatifs d'une population.

01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



Techniques statistiques avancées : Analyse des Z-scores

Explorer la Variabilité des Données :

Les Z-scores indiquent de combien d'écart-types un point de données s'éloigne de la moyenne.

En utilisant la bibliothèque NumPy de Python, on peut générer des données aléatoires suivant une distribution normale avec une moyenne de 50 et un écart-type de 10. Ensuite, les Z-scores pour chaque point de données sont calculés en utilisant la formule **(données - moyenne) / écart-type**.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# Generate random data (replace this with your own dataset)
data = np.random.normal(loc=50, scale=10, size=1000)
# Calculate Z-scores
z_scores = (data - np.mean(data)) / np.std(data)
```

01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers

Techniques statistiques avancées : Analyse des Z-scores

Explorer la Variabilité des Données :

Deux histogrammes ont été créés pour visualiser les données.

L'un montre la distribution des valeurs initiales sur l'axe des x,

L'autre présente la distribution des Z-scores sur l'axe des x..

Interprétation :

L'histogramme des Z-scores offre un aperçu de la variation des points de données par rapport à la moyenne.

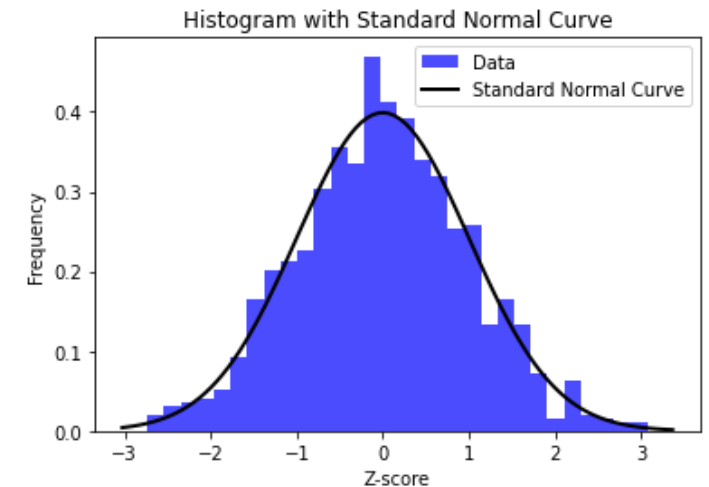
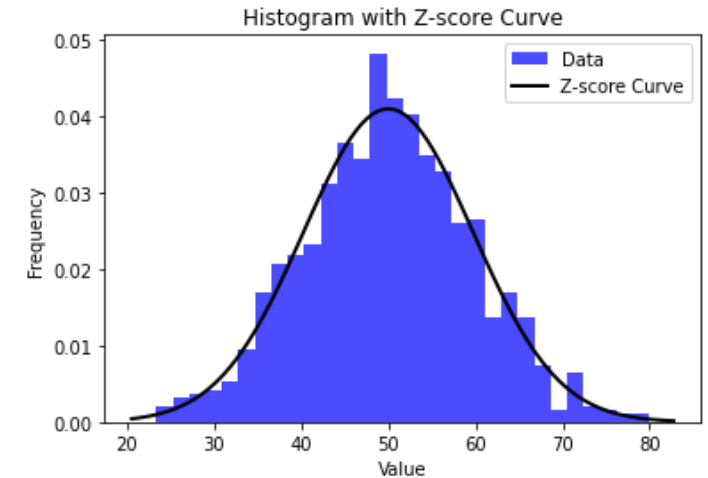
Par exemple, un Z-score de 2 indique une déviation de 20 par rapport à la moyenne.

L'application d'un seuil de 3 ou 4 pourrait être utilisée pour identifier et exclure les valeurs aberrantes qui s'éloignent significativement de la moyenne dans cette distribution normale.

Remarques :



Le Z-score diffère des valeurs brutes, avec une variation toujours à une échelle plus petite. On peut également utiliser le terme standardiser pour décrire ce processus, suggérant parfois le remplacement de certaines valeurs par leur version standardisée.



01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



La Standardisation et la Normalisation : Différences Théoriques et Utilisations Pratiques

Standardisation :

Théorie :

But : Transformer les données pour avoir une moyenne de 0 et un écart-type de 1.

Effet : Les données standardisées ont une variance unitaire et peuvent prendre n'importe quelle valeur. (positive et négatif)

Utilisations Pratiques :

1. Détection des Valeurs Aberrantes : Utilisation des z-scores pour identifier les outliers (valeurs au-delà de ± 3).

2. Algorithmes de Machine Learning :

Régression Linéaire et Logistique : Meilleure performance avec des données standardisées.

Analyse en Composantes Principales (ACP) : Requiert des caractéristiques sur la même échelle.

Clustering K-Means : Fonctionne mieux avec des caractéristiques standardisées pour éviter la domination de certaines caractéristiques.

- Après l'application de la **Standardisation** sur vos données normales, votre distribution sera comme ça :



01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



La Standardisation et la Normalisation : Différences Théoriques et Utilisations Pratiques

Normalisation :

Théorie :

But : Échelonner les données pour qu'elles s'inscrivent dans une plage spécifique, généralement $[0, 1]$.

Effet : Les données normalisées sont limitées à une plage fixe sans changer la forme de la distribution.

Utilisations Pratiques :

Réseaux de Neurones : Convergence plus rapide avec des entrées normalisées.

K-Nearest Neighbors (KNN) : Distance calculée de manière équitable entre les caractéristiques.

Visualisation des Données : Améliore l'interprétabilité des graphiques en maintenant une plage cohérente.

- Après l'application de la **Normalisation** vos données ressembleront à ceci :

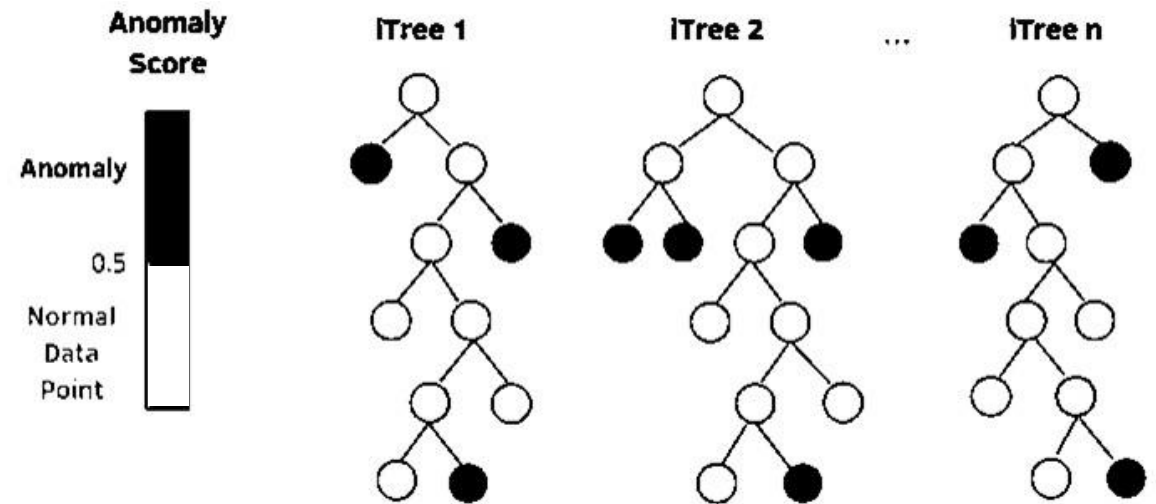
$[0, \dots, 1]$

Identification par les techniques statistiques avancées : Isolation Forest (Intrusion detection algorithm)

Isolation Forest est un algorithme de ML détecte les anomalies à l'aide d'**arbres binaires**. L'algorithme a une complexité temporelle linéaire (À mesure que les données augmentent, le temps de détection des anomalies avec Isolation Forest augmente linéairement) et une faible exigence en mémoire, adaptée aux données de grand volume. Il détecte les outliers en mesurant la facilité avec laquelle chaque point de données peut être isolé dans un arbre de décision.

Fonctionnement de l'Isolation Forest :

L'algorithme fonctionne en isolant rapidement les anomalies dans des arbres de décision. Les anomalies nécessitent moins de divisions pour être isolées, car elles ont des caractéristiques distinctes qui les rendent plus faciles à séparer des valeurs normales.



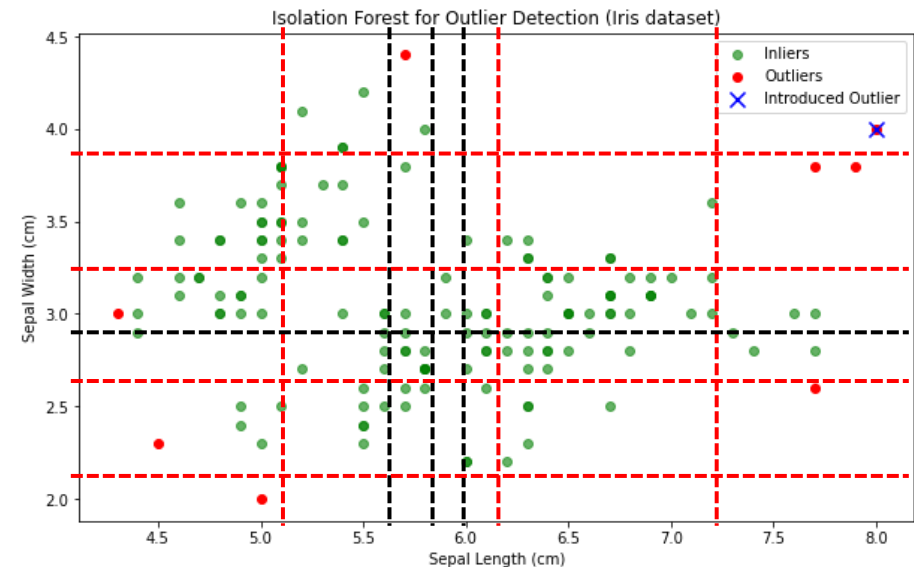
Identification par les techniques statistiques avancées : Isolation Forest (Intrusion detection algorithm)

Isolation Forest est un algorithme de ML basé sur des arbres de décision qui isole les anomalies plus efficacement que les méthodes traditionnelles. Il détecte les outliers en mesurant la facilité avec laquelle chaque point de données peut être isolé dans un arbre de décision.

Un Outlier sera détectée en nécessitant moins d'isolations que les valeurs typiques.

Comparé à d'autres méthodes, l'Isolation Forest est souvent plus rapide et nécessite moins de données pour détecter les anomalies.

Il est particulièrement adapté aux ensembles de données de grande dimension.



Introduced Outlier: ajouté intentionnellement pour tester la détection d'anomalies

Remarques :



Il est essentiel de comprendre les caractéristiques spécifiques de votre ensemble de données et de comparer différentes méthodes en fonction de vos besoins avant de choisir l'approche de détection des anomalies la plus

01 – Maîtriser les mesures de dispersion et de position

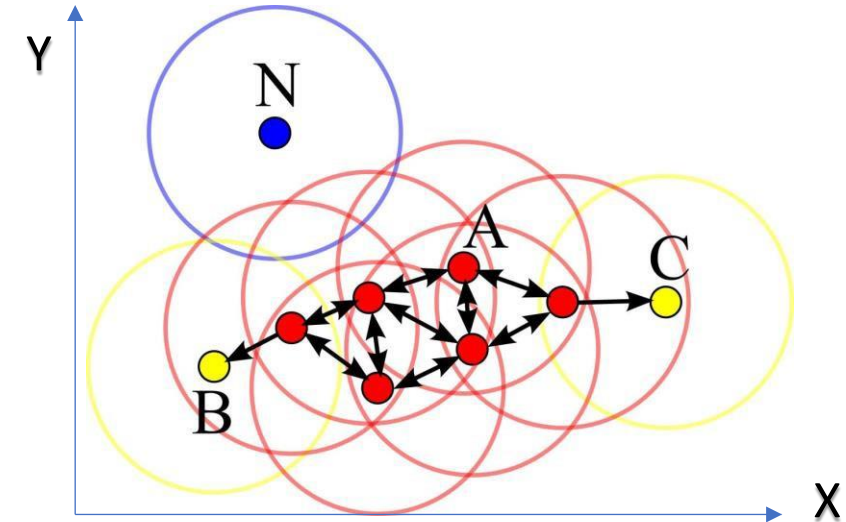
Analyse des Outliers



Techniques statistiques avancées : DBSCAN Algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

est un **algorithme de clustering basé sur la densité**. Il identifie les zones denses de points de données en se basant sur le nombre de points voisins « **Neighbours** » dans une région donnée. Les points qui ne sont pas atteints par suffisamment de voisins sont considérés comme du bruit. DBSCAN est efficace pour détecter des groupes de formes arbitraires et s'**adapte bien à différentes densités locales dans l'ensemble de données**.



A : Core points (points centraux)
B, C : Density-connected (connectés par densité)
N : Noise (bruit) => **Outliere**

Remarques :



DBSCAN est utilisé dans des situations complexes où **plusieurs colonnes ou dimensions sont prises en compte simultanément** pour construire de **nombreux clusters**. Cette approche diffère des méthodes précédentes qui détectent principalement les valeurs aberrantes en se basant sur les données d'une seule dimension.

DBSCAN est un algorithme de regroupement dont le but principal est d'identifier les régions denses dans les données. Bien qu'il puisse être utilisé pour détecter les valeurs aberrantes (Outliers).

01 – Maîtriser les mesures de dispersion et de position

Analyse des Outliers



Gestion et traitement des Outliers : Consielles

- Gérer les outliers est essentiel pour garantir la précision des analyses de données. Pour ce faire, diverses méthodes peuvent être utilisées, telles que la règle de IQR, les algorithmes DBSCAN et Isolation Forest.
- Les outliers ont une signification statistique, pouvant signaler des erreurs ou des événements spéciaux dans les données. Leur présence peut influencer considérablement sur les résultats et les conclusions tirées des analyses.
- Les Z-scores sont souvent utilisés pour identifier ces valeurs aberrantes.
- Enfin, la gestion des outliers est un processus continu qui nécessite une surveillance constante pour maintenir la qualité des analyses et des résultats.

01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification par les techniques statistiques avancées : DBSCAN multi-clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Pour cette illustration de DBSCAN, voici quelques points importants :

1. Paramètres Importants :

eps (epsilon) : Rayon maximal pour qu'un point soit considéré comme voisin.

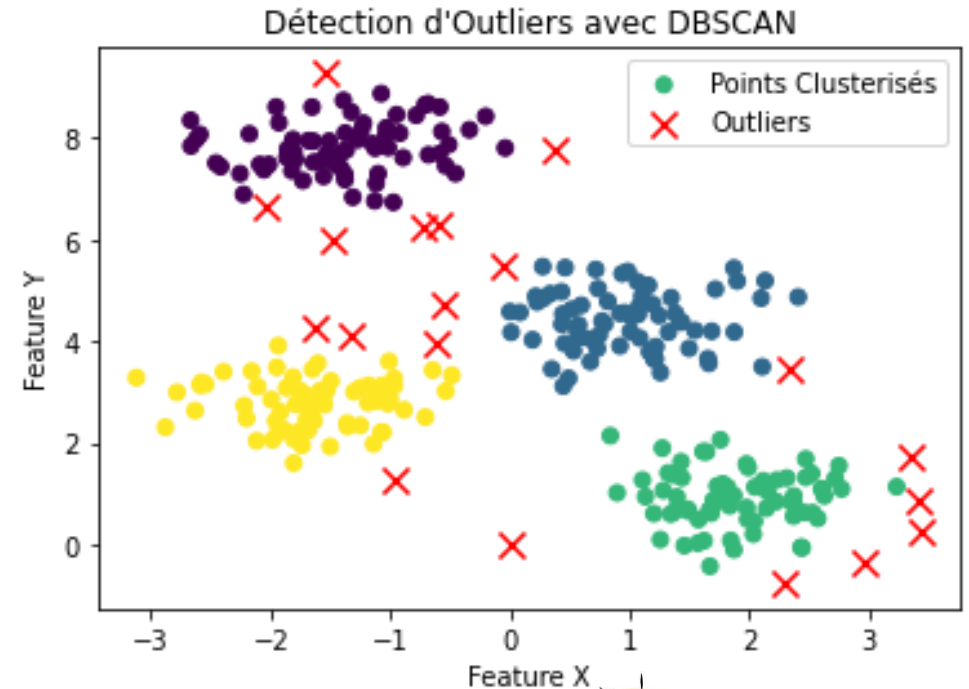
min_samples : Nombre minimal de points dans un cluster.

2. Commandes Essentielles en Python :

- `DBSCAN(eps=0.5, min_samples=5)` # Initialisation de l'algorithme DBSCAN
- `fit_predict(data)` # Prédire les clusters
- `clustered_data = data [labels != -1]` # Séparation des données clusterisées
- `outlier_data = data [labels == -1]` # Séparation des outliers fonction des labels

3. Visualisation avec Matplotlib :

- Scatter plot avec des points clusterisés colorés selon les clusters identifiés.
- Outliers marqués en rouge avec le symbole 'x'.
- Affichage du titre et des axes.



Remarques :



Des implémentations de DBSCAN peuvent être trouvées dans scikit-learn (R et Python).

Importation : `from sklearn.cluster import DBSCAN`

epsilon ou eps : le rayon autour de chaque point

01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification par les techniques statistiques avancées : DBSCAN multi-clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Solution :

Importer les bibliothèques et générer une distribution de données :



Remarques :



En cas de difficulté de compréhension de certains éléments. Il est recommandé de revenir à cette application après avoir avancé dans le cours, notamment sur la génération de distributions et la visualisation par scatter plot. Une autre application de DBSCAN sur des données réelles (tips dataset) est fournie sur la partie TP

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
from sklearn.datasets import make_blobs

# Générer des données de test avec quelques clusters et outliers
data, _ = make_blobs(n_samples=300, centers=4, cluster_std=0.60, random_state=0)
outlier = np.array([[0, 0]])
data_with_outlier = np.concatenate([data,
```

01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification par les techniques statistiques avancées : DBSCAN multi-clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Plus sur la génération de distribution de données avec `make_blobs` :

`make_blobs` est une fonction de la bibliothèque `sklearn.datasets` en Python, utilisée pour générer des ensembles de données synthétiques pour des exemples et des tests. Plus précisément, elle crée des blobs de points de données, ce qui est utile pour les démonstrations de techniques de clustering ou d'autres algorithmes de machine Learning.

Voici une brève description de ses principaux paramètres :

`n_samples` : Nombre total de points générés. Vous pouvez également passer une liste pour spécifier le nombre de points pour chaque

`cluster.centers` : Nombre de centres de clusters à générer, ou les coordonnées des centres de clusters.

`n_features` : Nombre de caractéristiques pour chaque point.

`cluster_std` : L'écart type des clusters.

`center_box` : Les limites des valeurs pour les centres des clusters.

`shuffle` : Indique si les échantillons doivent être mélangés.

`random_state` : Contrôle le générateur de nombres aléatoires pour la reproductibilité des résultats.

01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification par les techniques statistiques avancées : DBSCAN multi-clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Solution : suit

C'est notre section dédiée de l'application pour utiliser DBSCAN dans la détection des valeurs aberrantes. Il suffit d'appliquer cette méthode pour identifier les outliers :

Application de DBSCAN : Le modèle DBSCAN est configuré avec des paramètres spécifiques (eps=0.5 et min_samples=5) et appliqué aux données contenant des outliers.

```
# Appliquer DBSCAN avec des paramètres spécifiques
dbscan = DBSCAN(eps=0.5, min_samples=5)
labels = dbscan.fit_predict(data_with_outlier)

# Séparer les données clusterisées et les outliers
clustered_data = data_with_outlier[labels != -1]
outlier_data = data_with_outlier[labels == -1]
```

Séparation des données : Les données sont ensuite divisées en deux groupes : les données clusterisées (celles qui ne sont pas des outliers) et les outliers (données avec le label -1).

01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification par les techniques statistiques avancées : DBSCAN multi-clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Solution de la Visualisation avec Matplotlib :

Signifie que la colormap "viridis" est utilisée pour colorer les points clusterisés.

Spécifie que les couleurs des points sont déterminées par les labels des clusters, en excluant les outliers (qui ont un label de -1)

spécifient les coordonnées des points sur les axes X et Y respectivement.

```
# Afficher les résultats avec un scatter plot
plt.scatter(clustered_data[:, 0], clustered_data[:, 1], c=labels[labels != -1], cmap='viridis',
            label='Points Clusterisés')
plt.scatter(outlier_data[:, 0], outlier_data[:, 1], c='red', marker='x', s=100, label='Outliers')

plt.title('Détection d\'Outliers avec DBSCAN')
plt.xlabel('Feature X')
plt.ylabel('Feature Y')
plt.legend()
plt.show()
```



La fonction `plt.scatter()` attend les coordonnées X et Y comme ses premiers deux arguments positionnels, et non comme des arguments nommés X et Y.

Ce script visualise la séparation des points en clusters et les outliers détectés par l'algorithme DBSCAN, en utilisant deux caractéristiques des données pour les axes X et Y.

01 – Maîtriser les mesures de dispersion et de position

Application pratique avec Python



Identification par les techniques statistiques avancées : DBSCAN multi-clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Description des éléments de visualisation :

Voici une explication des différents éléments du graphique :

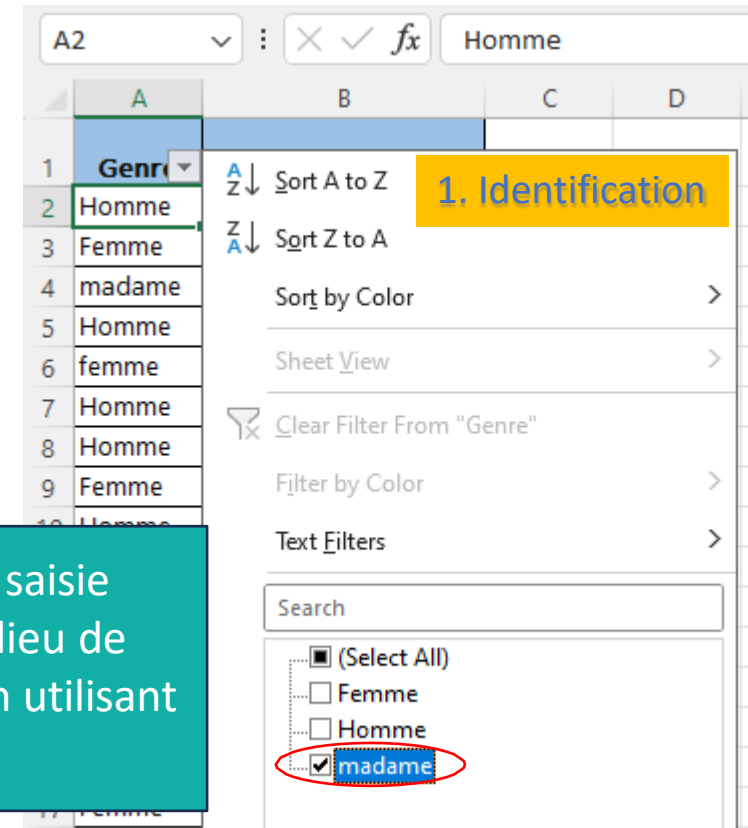
- **Points Clusterisés (Scatter Plot Vert) :** Les points sont affichés à l'aide d'un scatter plot, où chaque point représente une observation dans les données. Les couleurs des points (obtenues à l'aide de la palette de couleurs 'viridis') indiquent les différents clusters auxquels ces points appartiennent. Seuls les points appartenant à des clusters sont colorés.
- **Outliers (Scatter Plot Rouge) :** Les outliers, ou points aberrants, sont affichés en rouge avec un marqueur 'x'. Ces points ont été identifiés par l'algorithme DBSCAN comme ne faisant partie d'aucun cluster (label -1). Axes :
- **Feature X :** Représente la première caractéristique ou dimension des données.
- **Feature Y :** Représente la deuxième caractéristique ou dimension des données.
- **Titre du Graphique :** "Détection d'Outliers avec DBSCAN", indique que le graphique illustre les résultats de la détection d'outliers utilisant l'algorithme DBSCAN.
- **Légende :** La légende identifie les différents groupes de points : "Points Clusterisés" : Points appartenant à des clusters. "Outliers" : Points considérés comme des outliers.

01 – Maitriser les mesures de dispersion et de position

Application pratique avec Excel

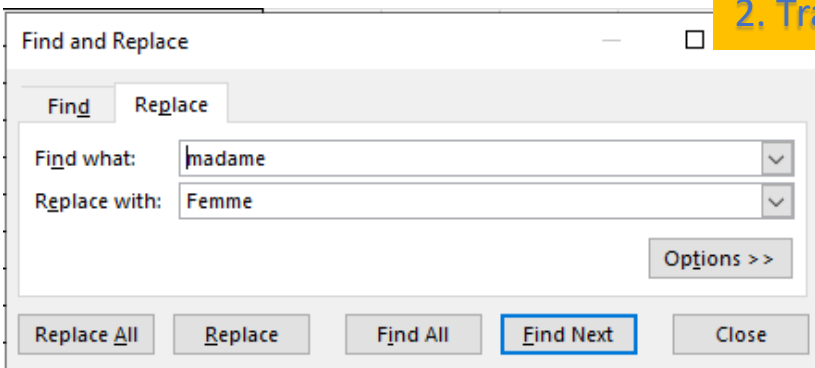
Identification et traitement des Outliers : Visualisation et traitement utilisent Excel

Malgré l'accès important aux outils statistiques et analytiques, dans de nombreuses situations professionnelles, utiliser des solutions simples et directes comme le Filtre d'Excel pour repérer les valeurs aberrantes et les traiter peut être une méthode valide et efficace, surtout avec l'existence d'un petit ensemble de données collectées directement auprès des utilisateurs, où une formule ou une feuille Excel.



	A	B	C	D
1	Genre			
2	Homme			
3	Femme			
4	madame			
5	Homme			
6	femme			
7	Homme			
8	Homme			
9	Femme			
10	Homme			

1. Identification



Find and Replace

Find what: madame

Replace with: Femme

Options >>

Replace All Replace Find All Find Next Close

2. Traitement

Identifier une faute de saisie (comme "madame" au lieu de "Femme") et la corriger en utilisant **Replace**

Ce type d'erreurs souvent lors du traitement de données provenant de différentes sources ou de Big Data. Normalement, de telles erreurs doivent être corrigées au cours du processus ETIL.

CHAPITRE 3

ASSIMILER LES PROBABILITÉS ET DISTRIBUTIONS AVANCÉES

Ce que vous allez apprendre dans ce chapitre :

- Explorer les notions avancées de probabilité
- Rappeler les distributions discrètes et continues
- Comprendre les fonctions de densité et de distribution
- Appliquer pratiquement avec Python/Excel



10 heures

