M202 - Maîtriser les techniques d'analyse de données

PARTIE 1: Approfondir les Bases de l'analyse des données

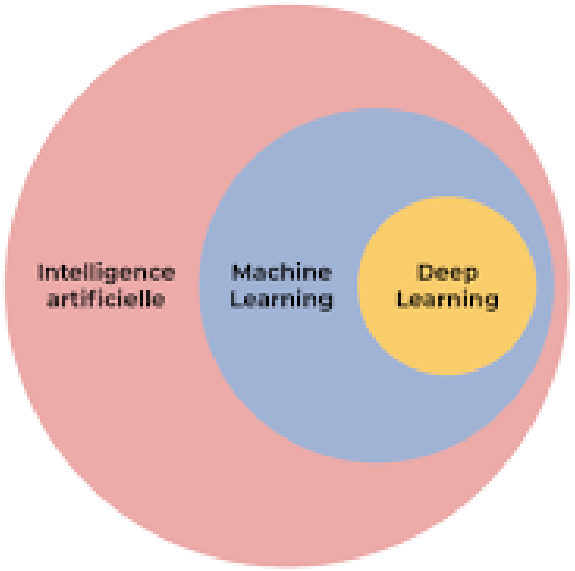
CHAPITRE 1: Revisiter les concepts clés en analyse de données

1. PRÉREQUIS de la compétence

2. Méthodes de nettoyage des données
   - Traitement des valeurs manquantes
   - Normalisation et standardisation
   - Codage des variables catégorielles

Prérequis de la compétence :

- Un bon niveau en Python.
- Un bon niveau en R.
- Se comprend mieux si vous maitrisez la compétence:
    - Introduction à l'apprentissage machine.
    - Introduction à l'apprentissage profond.



1. PRÉREQUIS de la compétence
2. Méthodes de nettoyage des données
   - Traitement des valeurs manquantes
   - Normalisation et standardisation
   - Codage des variables catégorielles

Différence entre valeur manquante et valeur absente:

Valeur manquante: la valeur était censée être présente, mais n'a pas été fournie pour diverses raisons.

valeur absente: la valeur n'était pas censée être présente, la variable n'est pas pertinente.

Nom	Role	Age	Note
YOUSEF	stagiaire	20	16.5
SARA	stagiaire	17	
MOHAMED	formateur	30	

Traitement des valeurs manquantes:

Dans le domaine de l'analyse des données, il est courant de faire face à des informations non capturées (données manquantes).

=> Quelles sont les causes qui entraînent des valeurs manquantes dans l'acquisition des données?

Traitement des valeurs manquantes:

- Causes des valeurs manquantes :
  - Erreurs de saisie : Erreurs humaines lors de la collecte de données.

Exemple: Lors d'une collecte de données pour une étude sur les habitudes alimentaires, un enquêteur peut accidentellement ignorer de saisir le poids d'un participant dans le formulaire. Cela crée une valeur manquante pour ce participant.

  - Problèmes techniques : Pannes de système ou erreurs de transmission de données.

Exemple: Une panne de serveur peut entraîner la perte de données pendant la transmission d'une enquête en ligne. Si certains participants remplissent le questionnaire, mais que leurs réponses ne sont pas enregistrées à cause de cette panne, cela crée des valeurs manquantes.

Traitement des valeurs manquantes:

- Causes des valeurs manquantes :
  - Non-réponse : Les participants d'une étude peuvent choisir de ne pas répondre à certaines questions.

Exemple : Dans une enquête sur la santé, les participants peuvent choisir de ne pas répondre à des questions sensibles, comme celles concernant leur consommation d'alcool ou leur état de santé mental. Cela résulte en valeurs manquantes pour ces questions spécifiques.

Traitement des valeurs manquantes:

- Impact des valeurs manquantes :
  - Biais dans les résultats : Si les valeurs manquantes ne sont pas gérées correctement, elles peuvent fausser les résultats de l'analyse.
  - Perte d'information : Plus il y a de valeurs manquantes, plus l'analyse peut être limitée.

Traitement des valeurs manquantes:

- Types de valeurs manquantes :
  - MCAR (Missing Completely At Random) : Valeurs manquantes complètement aléatoires :

Exemple : Lors d'une enquête sur la satisfaction client, certaines réponses sont perdues à cause d'un problème informatique aléatoire. Ces valeurs manquantes ne sont liées ni aux réponses des participants ni à leurs caractéristiques (âge, genre, etc.). Elles sont donc aléatoires.

==> Ces valeurs manquantes n'introduisent pas de biais dans les résultats, car leur absence est indépendante des autres variables.

Traitement des valeurs manquantes:

- Types de valeurs manquantes :
  - MAR (Missing At Random) : Valeurs manquantes aléatoires conditionnées :

Exemple : Dans une étude sur l'activité physique, les participants plus âgés peuvent être plus susceptibles de ne pas répondre aux questions sur la fréquence des exercices physiques. Les valeurs manquantes concernant la fréquence de l'exercice dépendent de l'âge des participants.

==> Puisque l'absence de réponse dépend d'une autre variable (l'âge), on peut utiliser cette information pour estimer ou imputer les valeurs manquantes.

Traitement des valeurs manquantes:

- Types de valeurs manquantes :

  - MNAR (Missing Not At Random) : Valeurs manquantes non aléatoires :

Exemple : Dans une étude sur la consommation de cigarettes, les personnes qui fument beaucoup peuvent ne pas répondre à des questions sur leur consommation de tabac. Les valeurs manquantes sont donc plus fréquentes chez les gros fumeurs. Ici, l'absence de réponse est directement liée à la variable (le nombre de cigarettes fumées par jour).

==> Les valeurs manquantes ne sont pas aléatoires et dépendent directement de la variable que l'on cherche à mesurer (la consommation de tabac). Cela peut introduire un biais important.
Sans corriger ce biais, l'analyse risque de sous-estimer la consommation moyenne de cigarettes dans l'étude.

Traitement des valeurs manquantes:

- Imputation des valeurs manquantes :

  - Imputation avec la Moyenne:

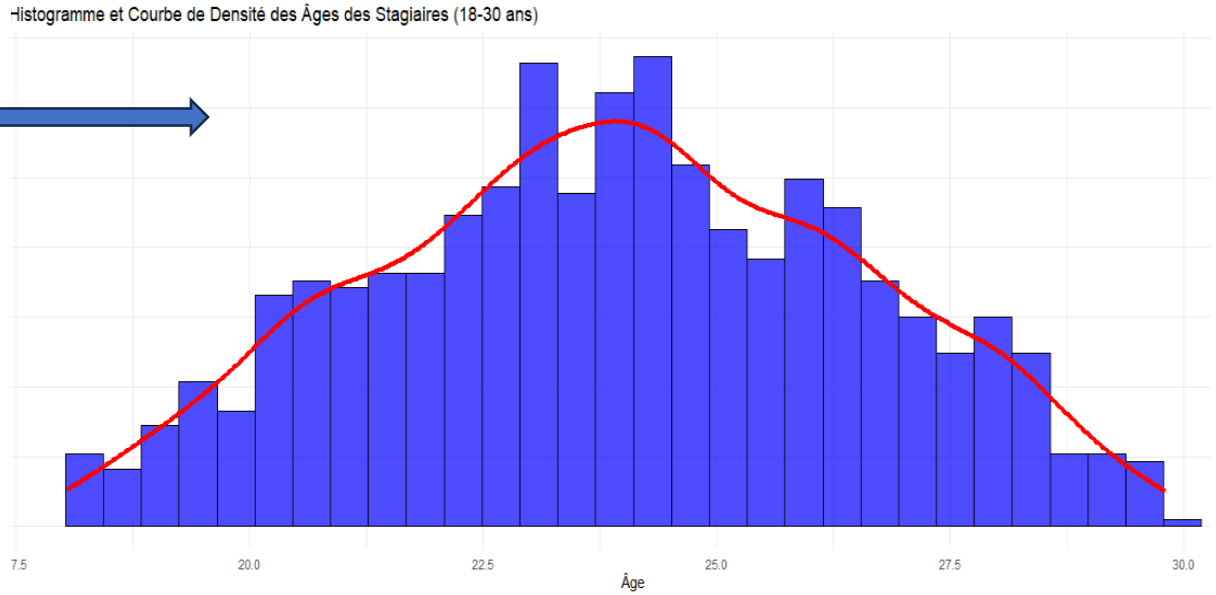
  Cas d'utilisation:

    - Distribution symétrique.
    - Nombre très faible de données manquantes < 15%. Sinon, il peut modifier la corrélation entre les variables et biaiser les modèles d'estimation.

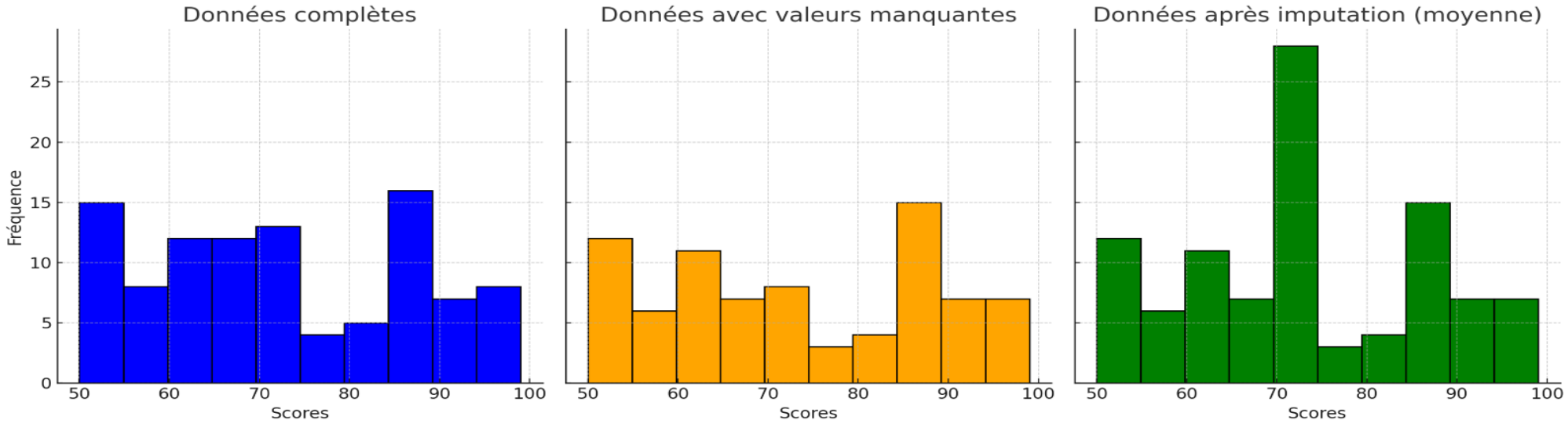
Exemple : distribution normale des âges des stagiaires.

- **Imputation avec la Moyenne:**

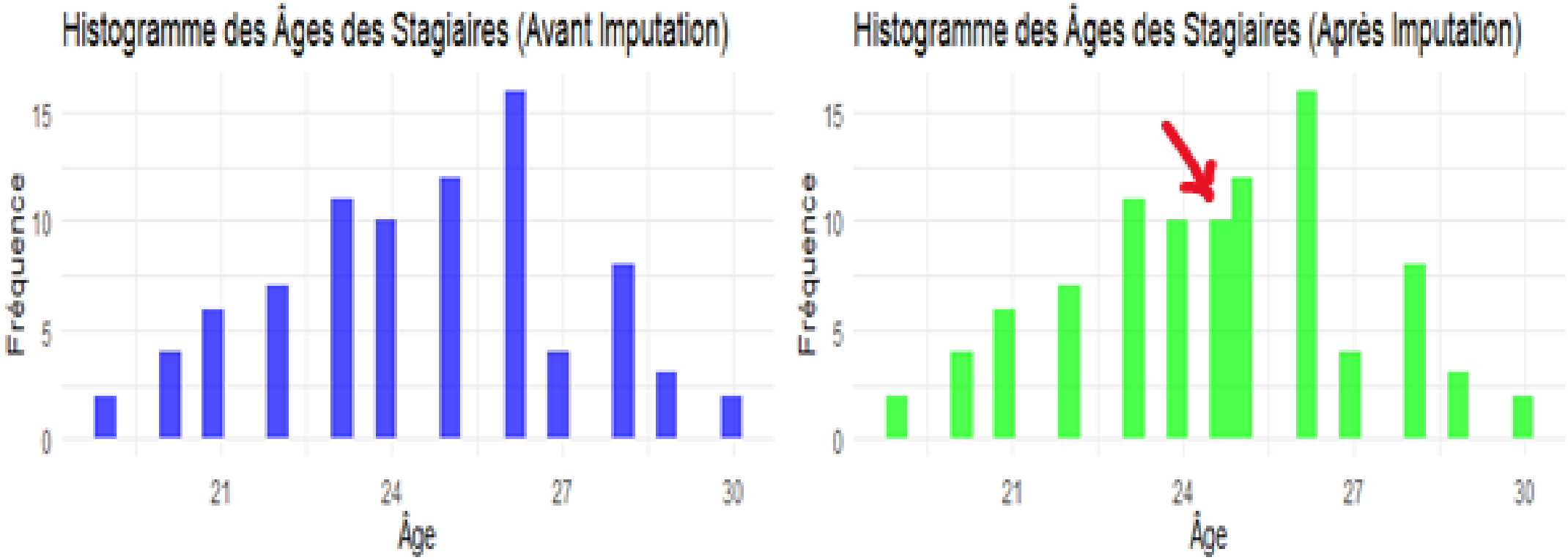
  - Distribution symétrique.



  - Effet sur une distribution non symétrique.



- **Imputation avec la Moyenne:**
  - Distribution symétrique.



Exemple : distribution normale des âges des stagiaires.

- **Imputation avec la Moyenne:**

  - Imputer les valeurs manquantes dans le jeu de données ci-dessous.

ville	age
casa	25
agadir	27
tanger	29
casa	26
fes	24
oujda	28
oujda	27
casa	30
rabat	31
tanger	32
nador	29
casa	28
fes	26
casa	**NaN**
rabat	**NaN**

Traitement des valeurs manquantes:

- Imputation des valeurs manquantes :

  - Imputation avec la Médiane :

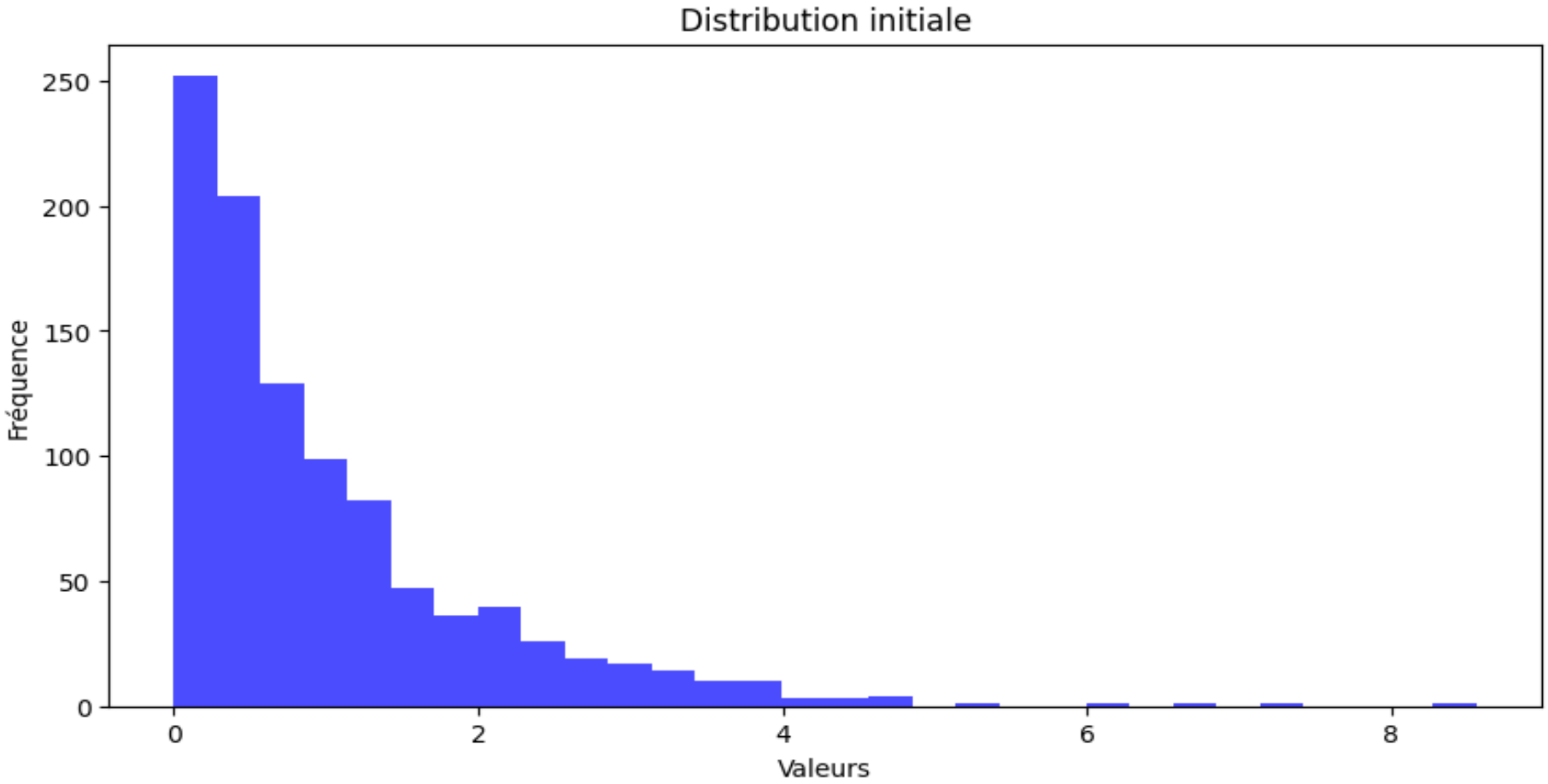
  Cas d'utilisation:

  - Distribution asymétrique (skewed distribution): c'est une distribution qui contient des valeurs extrêmes (outliers) qui peuvent affecter de manière significative la moyenne, mais pas la médiane.

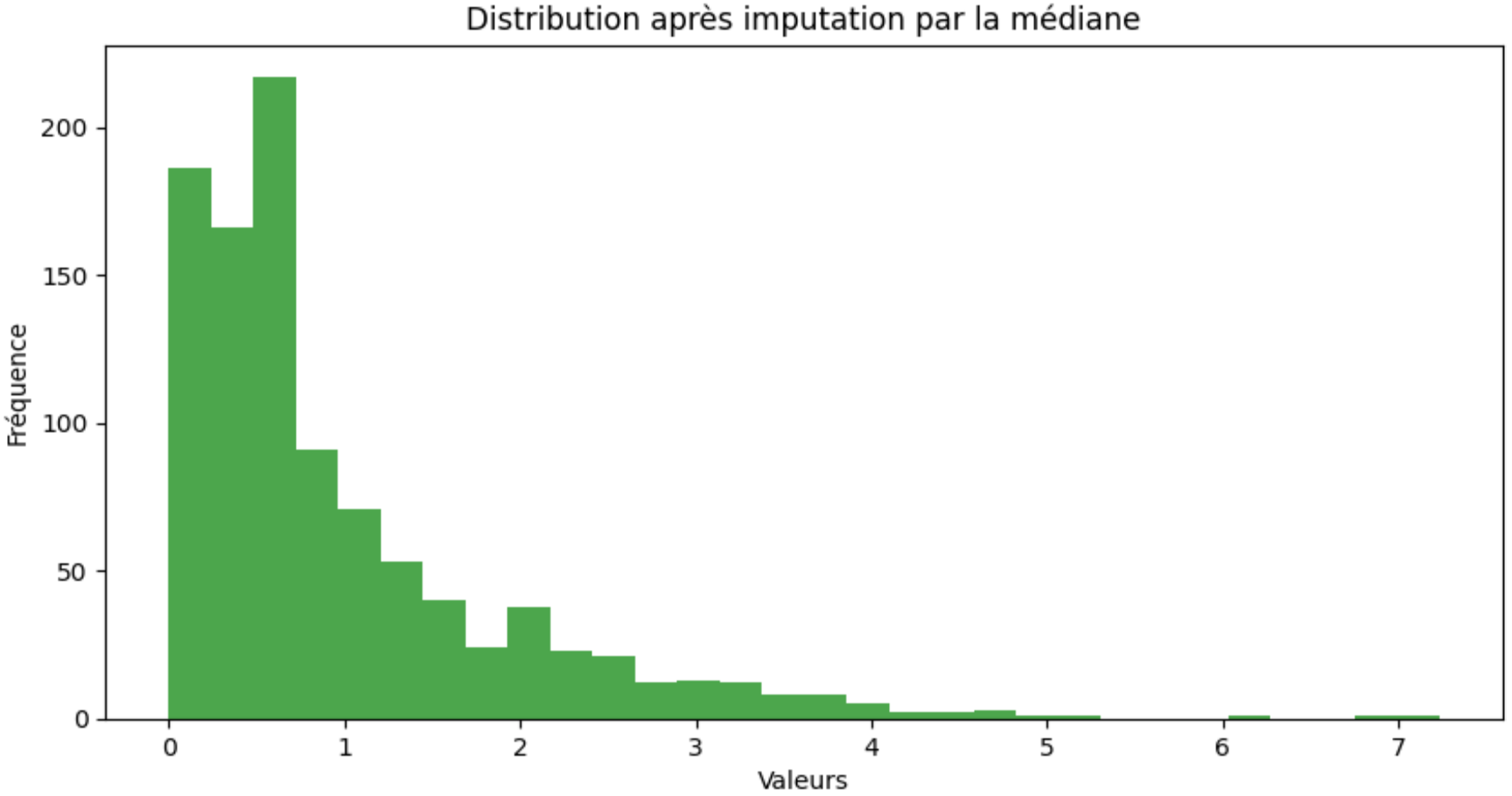
  - Nombre très faible de données manquantes < 15%. Sinon, il peut modifier la corrélation entre les variables et biaiser les modèles d'estimation.

- **Imputation avec la Médiane:**

  - Distribution asymétrique avant imputation.

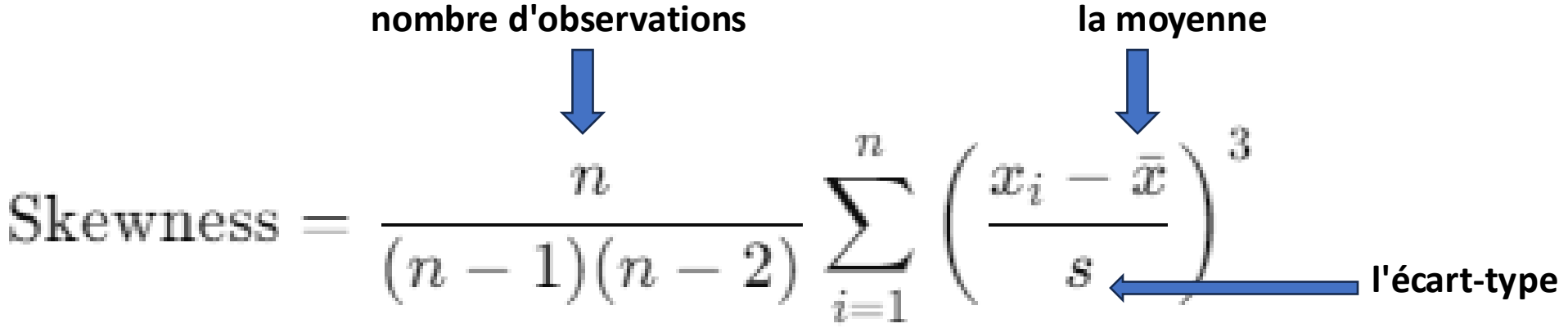


- **Imputation avec la Médiane:**
  - Distribution asymétrique après imputation.

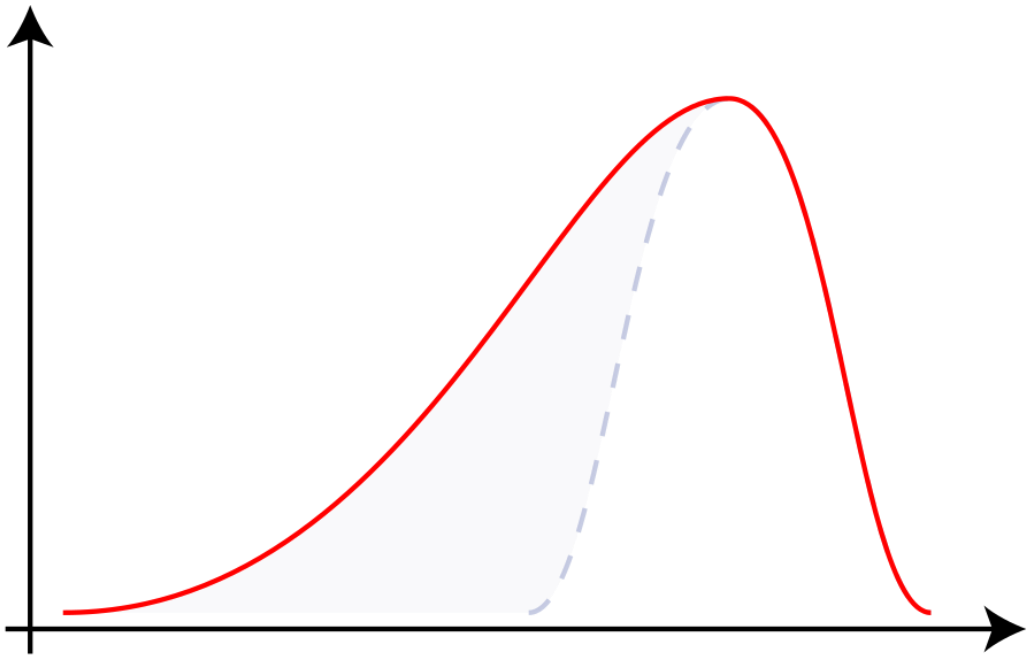


- Imputation avec la Médiane :

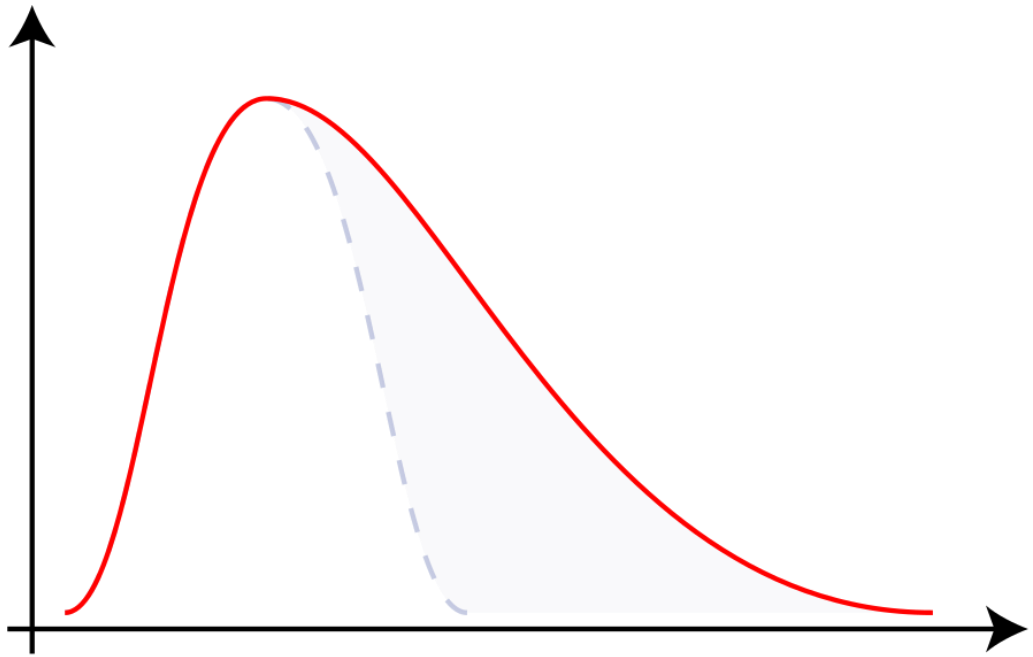
Le **skewness** d'une distribution peut être calculé avec la formule suivante



- Skewness > 0 : asymétrie positive.
- Skewness < 0 : asymétrie négative.
- Skewness ≈ 0 : distribution symétrique (comme une distribution normale).



Negative skew

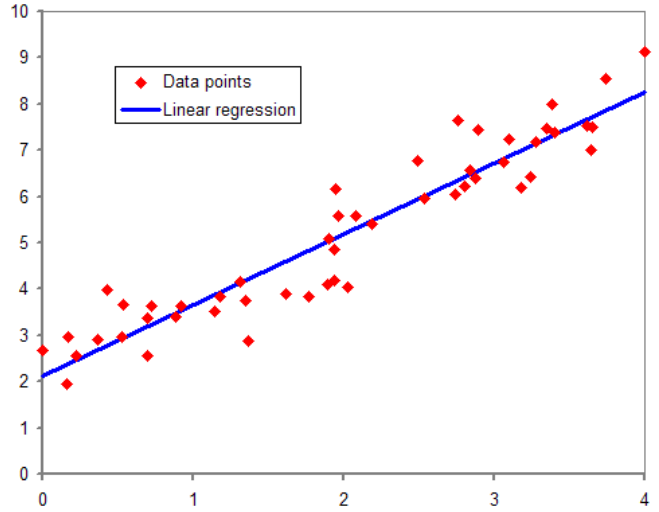


Positive skew

- **Imputation avec la Médiane:**
  - Exemple d'application:
    - Importer le fichier revenu_annuel.csv et imputer les valeurs manquantes.

- NP: si le nombre des valeurs manquantes dépasse ~15 % il faut utiliser d'autres méthodes d'imputation.

Traitement des valeurs manquantes:



- Imputation des valeurs manquantes :

  - Imputation avec la Régression :

  Cas d'utilisation:

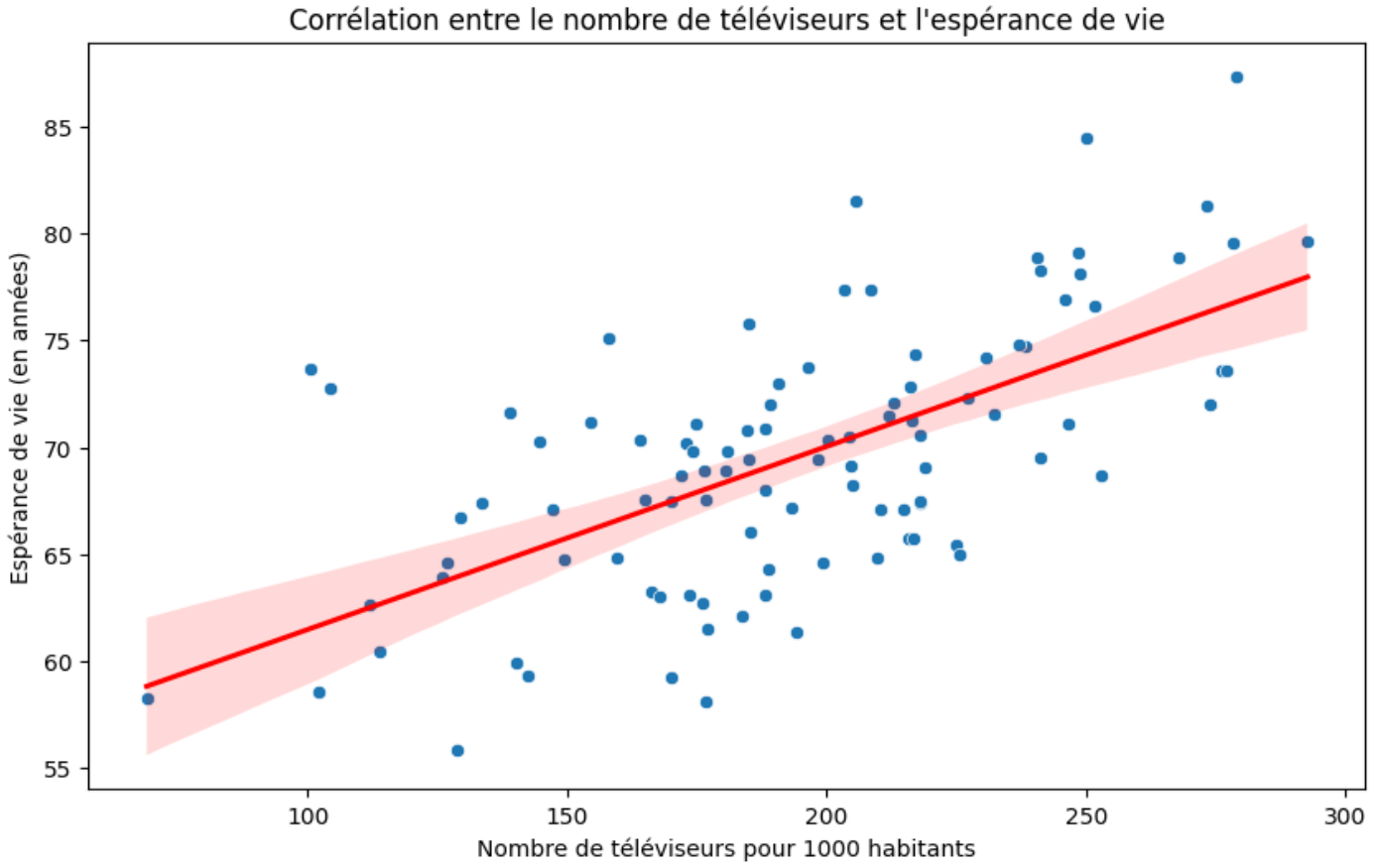
  - Existence de relations : Utilisez l'imputation par régression lorsque vous avez une ou plusieurs variables indépendantes (prédicteurs) qui sont corrélées avec la variable ayant des valeurs manquantes.

  - Si la proportion de valeurs manquantes est modérée (par exemple, moins de 30 % des données), l'imputation par régression peut fournir de bonnes estimations. Une proportion plus élevée de valeurs manquantes pourrait nécessiter d'autres approches.

- "La corrélation n'implique pas toujours la causalité"

- Imputation avec la Régression :

"La corrélation n'implique pas toujours la causalité":



- Même si une corrélation positive est visible sur le graphique, il est important de rappeler que cela n'implique pas nécessairement que posséder plus de téléviseurs augmente l'espérance de vie.

- Il s'agit plutôt d'une coïncidence basée sur d'autres facteurs, comme la richesse d'un pays.

Traitement des valeurs manquantes:

- Imputation avec la Régression :

Exemple d'application:

Taille (cm)	Poids(kg)	Age
170	65	30
160	58	25
180	**NaN**	40
175	75	35
165	**NaN**	28
185	**85**	42

- Type de valeurs manquantes: MAR (c'est-à-dire manquantes de manière aléatoire mais dépendant d'autres variables observées dans le jeu de données.)
- Nombre de valeurs manquantes vaut 33%.

==> **Etudier la corrélation entre les variables indépendantes (taille et age) et la variable poids**

Traitement des valeurs manquantes:

- Imputation avec la Régression :

Exemple d'application:

Taille (cm)	Poids(kg)	Age
170	65	30
160	58	25
180	**NaN**	40
175	75	35
185	**NaN**	50
161	**64**	26

- Type de valeurs manquantes: MAR (c'est-à-dire manquantes de manière aléatoire mais dépendant d'autres variables observées dans le jeu de données.)
- Nombre de valeurs manquantes vaut 33%.

==> **Etudier la corrélation entre les variables indépendantes (taille et age) et la variable poids**

- Imputation avec la Régression :

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

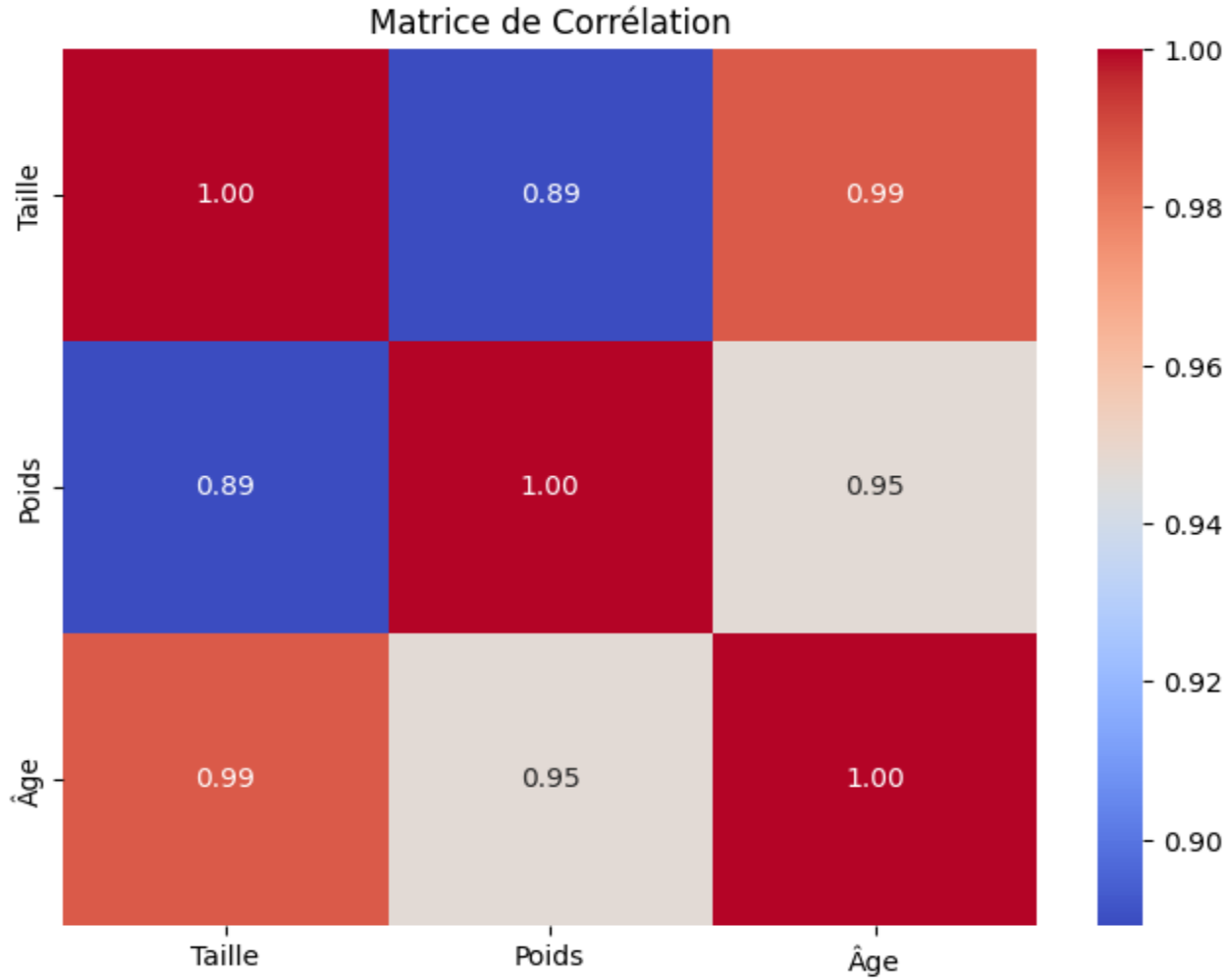
```
# représenter les données en dictionnaire
data = {
    'Taille': [170, 160, 180, 175, 185,161],
    'Poids': [65, 58, None, 75, None,64],
    'Âge': [30, 25, 40, 35, 45, 26]
}
```

```
# Création d'un DataFrame
df = pd.DataFrame(data)
```

	Taille	Poids	Âge
0	170	65.0	30
1	160	58.0	25
2	180	NaN	40
3	175	75.0	35
4	185	NaN	45
5	161	64.0	26

Taille (cm)	Poids(kg)	Age
170	65	30
160	58	25
180	**NaN**	40
175	75	35
185	**NaN**	45
161	**64**	26

- Imputation avec la Régression :
  - Calcul de la corrélation
  - Visualisation de la matrice de corrélation avec un heatmap

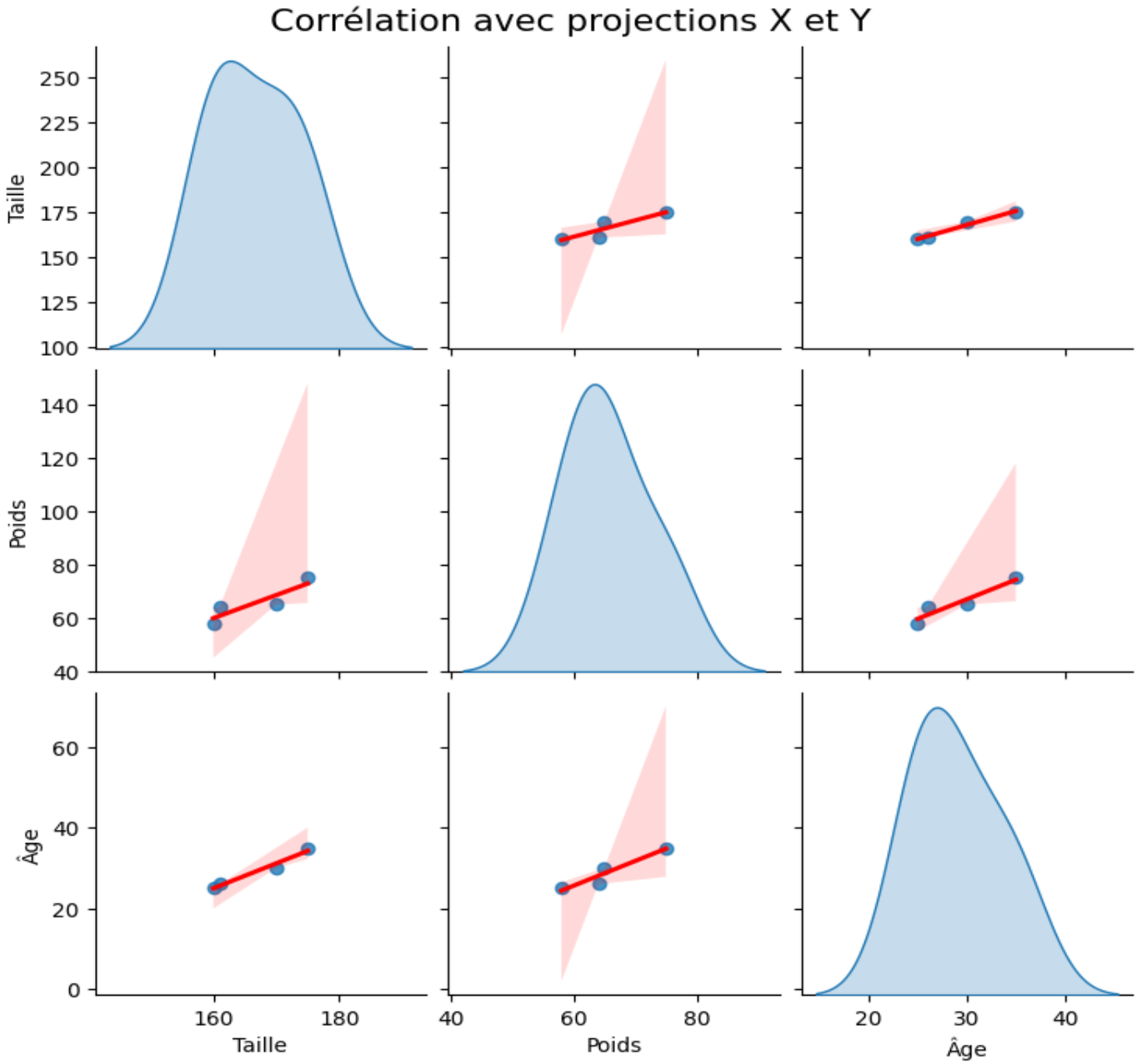


Taille (cm)	Poids(kg)	Age
170	65	30
160	58	25
180	**NaN**	40
175	75	35
185	**NaN**	45
161	**64**	26

- Imputation avec la Régression :
  - Création de de graphe en paire (Pairplot)

Taille (cm)	Poids(kg)	Age
170	65	30
160	58	25
180	**NaN**	40
175	75	35
185	**NaN**	45
161	**64**	26

=> On observe une corrélation linéaire entre le poids et (Taille, Age).



- Imputation avec la Régression :

- Les étapes d'imputation par la régression linéaire :

✓ Séparer les lignes avec et sans poids manquant:
train_data
test_data

Taille (cm)	Poids(kg)	Age
170	65	30
160	58	25
180	NaN	40
175	75	35
185	NaN	45
161	64	26

✓ Variables indépendantes et dépendante:
X_train --> train_data[['Taille', 'Âge']]
y_train --> train_data[['Poids']]

✓ Création et entraînement du modèle de régression:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

✓ Prédire les poids manquants:
```
X_test = test_data[['Taille', 'Âge']]
predictions = model.predict(X_test)
```

✓ Imputation des valeurs manquantes

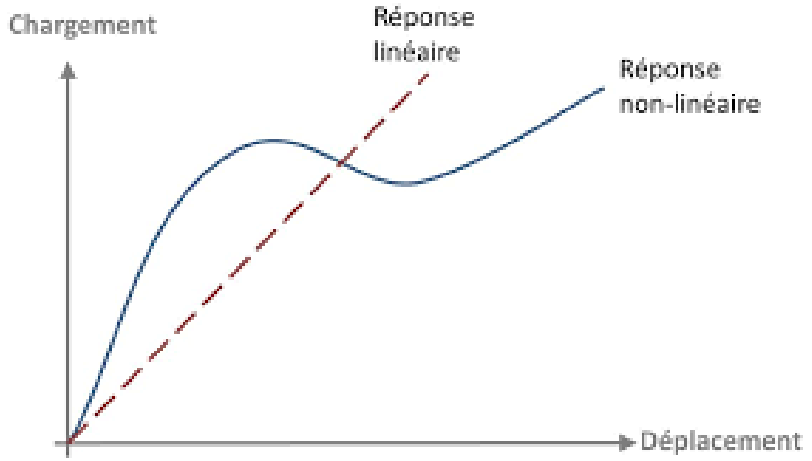
	Taille	Poids	Âge
0	170	65.000000	30
1	160	58.000000	25
2	180	86.153846	40
3	175	75.000000	35
4	185	96.923077	45
5	161	64.000000	26

Exemple d'application:

- Analyser les méthodes d'imputation à utiliser dans le jeu de données ci-dessous:

	prix_m2	age_batiment	distance_metro
0	3000.0	10.0	500
1	3200.0	15.0	700
2	3100.0	12.0	600
3	3500.0	20.0	800
4	3300.0	18.0	750
5	NaN	25.0	900
6	3400.0	NaN	650
7	3700.0	NaN	550
8	NaN	5.0	400

Traitement des valeurs manquantes:



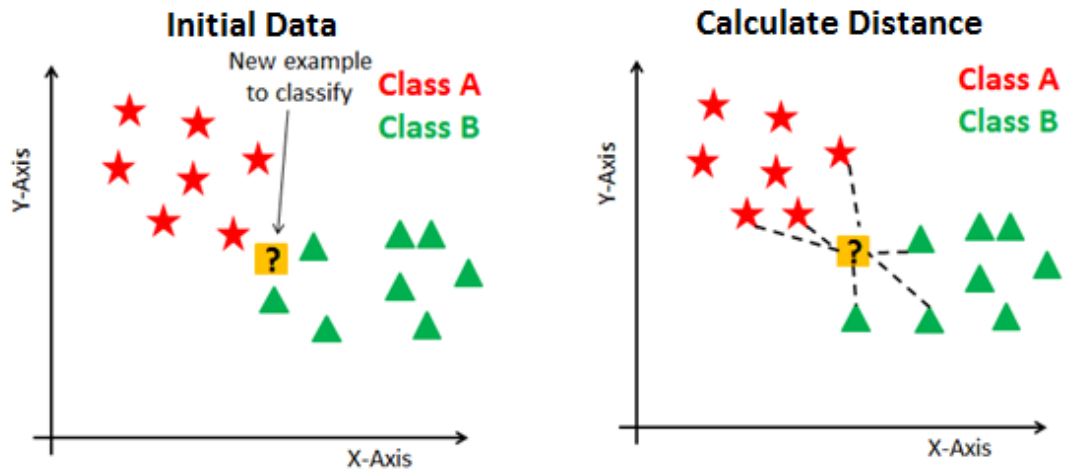
- Imputation des valeurs manquantes :

  - Imputation avec K-Nearest Neighbors (KNN) :

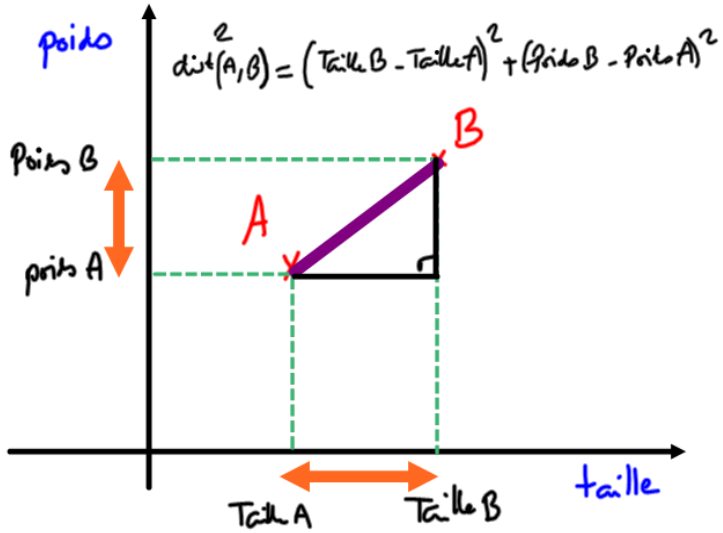
  Cas d'utilisation:

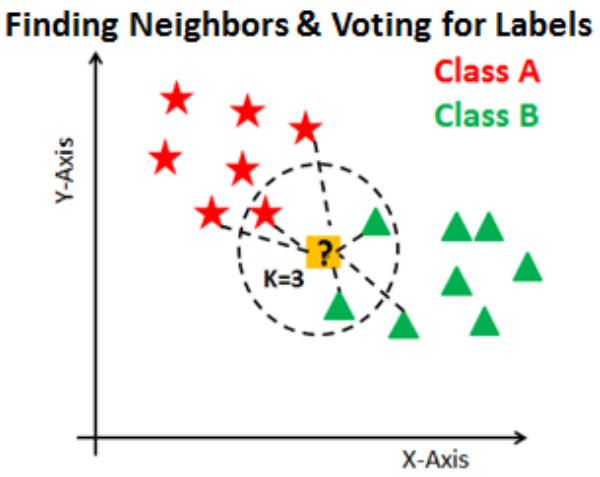
  - Il y a des relations complexes entre les variables : Si vous avez des variables qui sont liées de manière non linéaire ou complexe, KNN peut capturer ces relations sans avoir besoin de modélisation explicite.
  - Le dataset n'est pas trop grand : KNN peut être coûteux en termes de calcul, surtout pour de grands ensembles de données, car il nécessite de calculer les distances entre chaque observation avec des données manquantes et tous les autres points du dataset.
  - Si la proportion de valeurs manquantes est modérée (par exemple, moins de 30 % des données).

- Rappel: K-Nearest Neighbors (KNN) :



Distance euclidienne



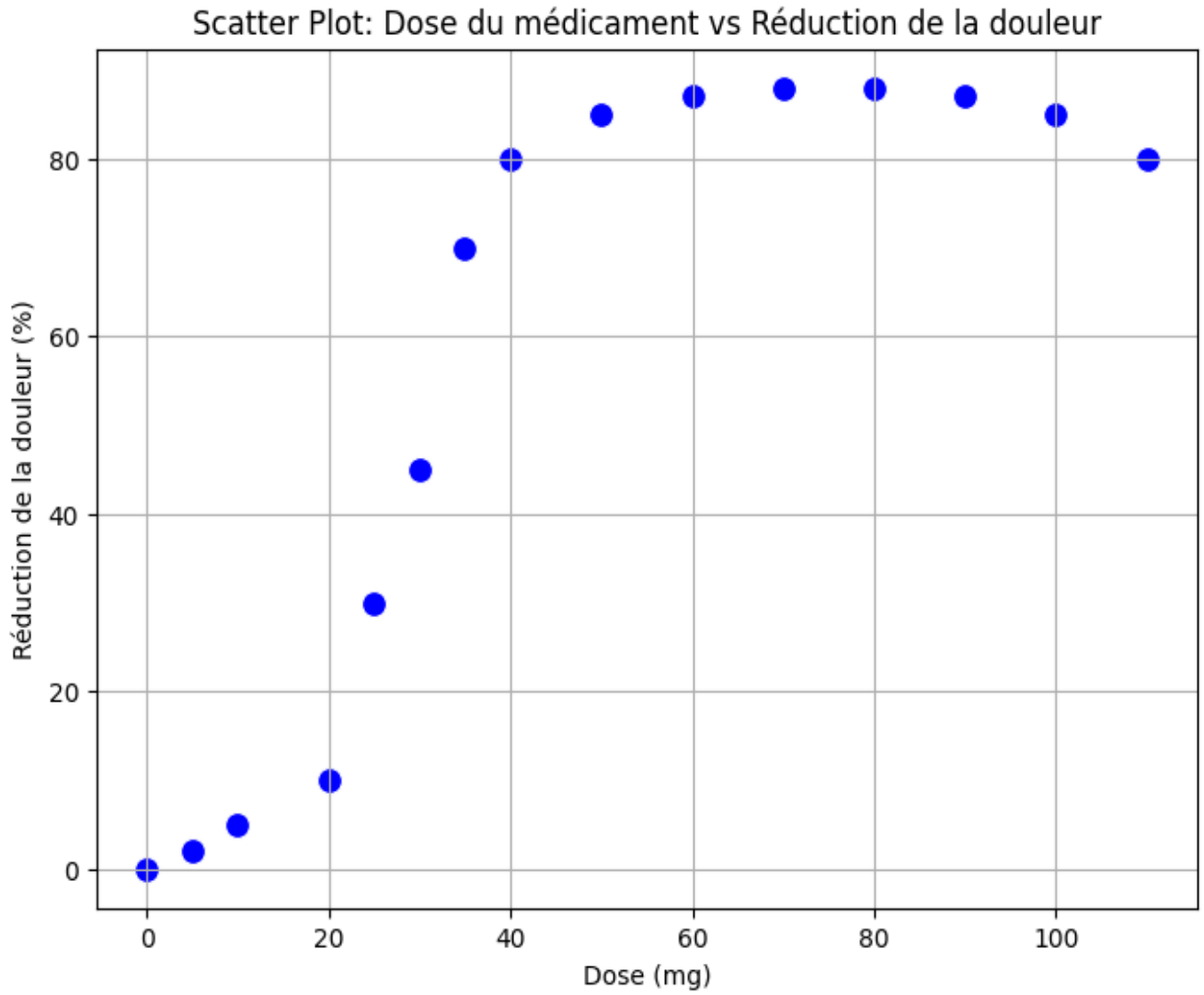


- Imputation avec KNN:

Exemple d'application (cas d'utilisation)

	Patient_ID	Dose (mg)	Réduction de la douleur (%)
0	1	0	0.0
1	2	5	2.0
2	3	10	5.0
3	4	17	NaN
4	5	20	10.0
5	6	22	NaN
6	7	25	30.0
7	8	30	45.0
8	9	35	70.0
9	10	40	80.0
10	11	50	85.0
11	12	60	87.0
12	13	70	88.0
13	14	80	88.0
14	15	90	87.0
15	16	100	85.0
16	17	110	80.0

Scatter Plot

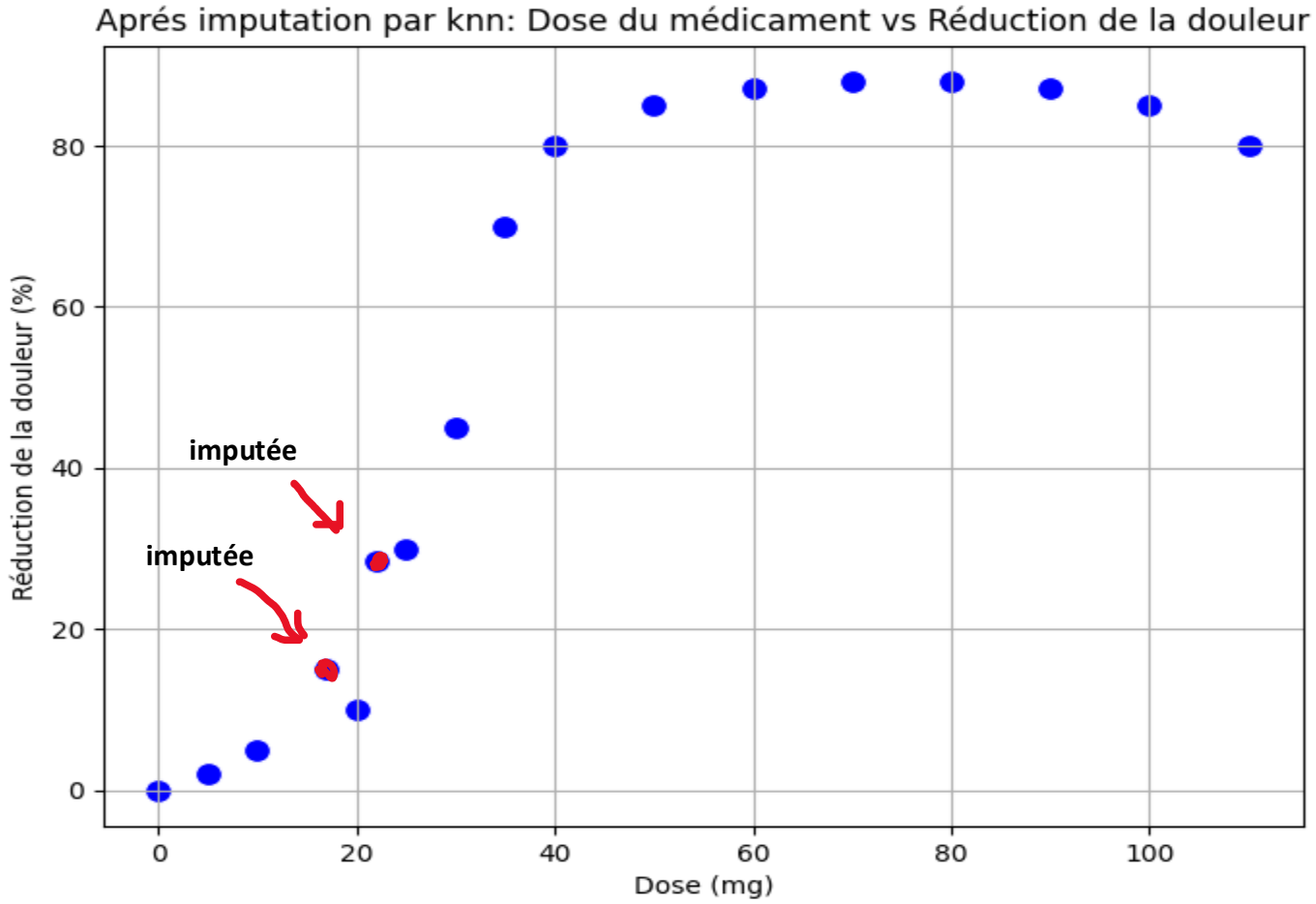


- **Exemple d'application:**:

- Analyser l'imputation à utiliser dans le jeu de données ci-dessous:
- Pour KNN utiliser la bibliothèque from sklearn.impute import KNNImputer

	Patient_ID	Dose (mg)	Réduction de la douleur (%)
0	1	0	0.0
1	2	5	2.0
2	3	10	5.0
3	4	17	NaN
4	5	20	10.0
5	6	22	NaN
6	7	25	30.0
7	8	30	45.0
8	9	35	70.0
9	10	40	80.0
10	11	50	85.0
11	12	60	87.0
12	13	70	88.0
13	14	80	88.0
14	15	90	87.0
15	16	100	85.0
16	17	110	80.0

Scatter Plot



1. PRÉREQUIS de la compétence
2. Méthodes de nettoyage des données
   - Traitement des valeurs manquantes
   - Normalisation et standardisation
   - Codage des variables catégorielles

Normalisation:

- Transformer les caractéristiques (features) pour les amener à des échelles comparables.
- lorsque des algorithmes basés sur la distance sont utilisés (KNN, k-means, etc.).
- Perte d'information sur la distribution normale.
- Redimensionner les valeurs des variables pour qu'elles soient comprises dans une certaine plage, généralement [0, 1]

Formule de la normalisation : $X' = (X - X_{min}) / (X_{max} - X_{min})$.

Normalisation:

- Implémentation:

```
from sklearn.preprocessing import MinMaxScaler
import numpy as np

# Exemple de données
data = np.array([[10, 2], [20, 3], [30, 5]])

# Appliquer la normalisation (Min-Max scaling)
scaler = MinMaxScaler()
normalized_data = scaler.fit_transform(data)

print(normalized_data)
```

Standardisation:

- Transformer les caractéristiques pour qu'elles aient une moyenne de 0 et un écart-type de 1.
- approprié lorsque les données suivent une distribution normale.
- souvent utilisée pour les algorithmes ML comme la régression linéaire, les modèles bayésiens, réseaux de neurones.

Formule de la standardisation : $X' = (X - \mu) / \sigma$. où μ est la moyenne et σ est l'écart-type.

Standardisation:

- Implémentation:

```
from sklearn.preprocessing import StandardScaler
import numpy as np

# Exemple de données
data = np.array([[10, 2], [20, 3], [30, 5]])

# Appliquer la standardisation
scaler = StandardScaler()
standardized_data = scaler.fit_transform(data)
print("Données standardisées :\n", standardized_data)
```

Standardisation:

- Utiliser le jeu de données "Breast Cancer" de scikit-learn, et standardiser les données numériques.

```
from sklearn.datasets import load_breast_cancer
data = load_breast_cancer()
X = data.data
y = data.target
```

1. PRÉREQUIS de la compétence
2. Méthodes de nettoyage des données
   - Traitement des valeurs manquantes
   - Normalisation et standardisation
   - Codage des variables catégorielles

Type des variables catégorielles:

- **Une variable nominale**: une variable catégorielle sans ordre naturel entre les catégories. Les catégories sont distinctes, mais aucune n'est supérieure ou inférieure à une autre.

  Exemples :
  - Couleur : [Rouge, Bleu, Vert]
  - Ville : [Paris, Lyon, Marseille]
  - Genre : [Homme, Femme]

- **les variables ordinales**: Ce sont des variables catégorielles où les catégories ont un ordre naturel. Les catégories peuvent être classées selon une hiérarchie.

  Exemples :
  - Niveau d'éducation : [Baccalauréat, Licence, Master, Doctorat]
  - Taille de vêtements : [Petit, Moyen, Grand]
  - Satisfaction client : [Très insatisfait, Insatisfait, Neutre, Satisfait, Très satisfait]

Codage des variables nominales:

- **One-Hot Encoding** : une technique où chaque catégorie est transformée en une nouvelle colonne (ou plusieurs colonnes) avec des valeurs binaires (0 ou 1).
  - Couleur : [Rouge, Bleu, Vert, Rouge]

```
df_encoded = pd.get_dummies(df, columns=['Couleur'])
```

	Couleur_Bleu	Couleur_Rouge	Couleur_Vert
0	0	1	0
1	1	0	0
2	0	0	1
3	0	1	0

Codage des variables nominales:

" One-hot encoding peut générer de très nombreuses colonnes "
==> augemente la complexité du modèle + rendre les données difficiles à traiter.

- **Target encoding** :
  - une technique où chaque catégorie est remplacée par la moyenne de la colonne cible.
  - Grand nombre de catégories.

```
# Calculer la moyenne de la cible (Achat) pour chaque Marque
target_mean = df.groupby('Marque')['Achat'].mean()

# Appliquer le target encoding en remplaçant la colonne Marque
df['Marque_encoded'] = df['Marque'].map(target_mean)
```

	Client_ID	Marque	Achat
0	1	Marque_52	0
1	2	Marque_93	1
2	3	Marque_15	0
3	4	Marque_72	0
4	5	Marque_61	0
...	...	...	...
995	996	Marque_10	1
996	997	Marque_67	1
997	998	Marque_18	1
998	999	Marque_100	0
999	1000	Marque_86	0

Codage des variables ordinales:

- **Ordinal Encoding** : utilisé lorsque les catégories ont un ordre naturel.

	ID	Couleur	Taille	Type de voiture	Prix	Kilométrage
0	1	Rouge	Petit	SUV	25000	30000
1	2	Bleu	Moyen	Berline	30000	25000
2	3	Vert	Grand	Coupé	35000	15000
3	4	Rouge	Moyen	SUV	26000	20000
4	5	Bleu	Petit	Berline	29000	22000
5	6	Vert	Grand	SUV	40000	5000
6	7	Rouge	Petit	Coupé	27000	12000
7	8	Bleu	Moyen	Berline	31000	17000

```
# Appliquer le codage ordinal pour 'Taille'
ordinal_encoder = OrdinalEncoder(categories=[['Petit', 'Moyen', 'Grand']])
df_encoded['Taille_encoded'] = ordinal_encoder.fit_transform(df_encoded[['Taille']])
```