M202 - Maîtriser les techniques d'analyse de données

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

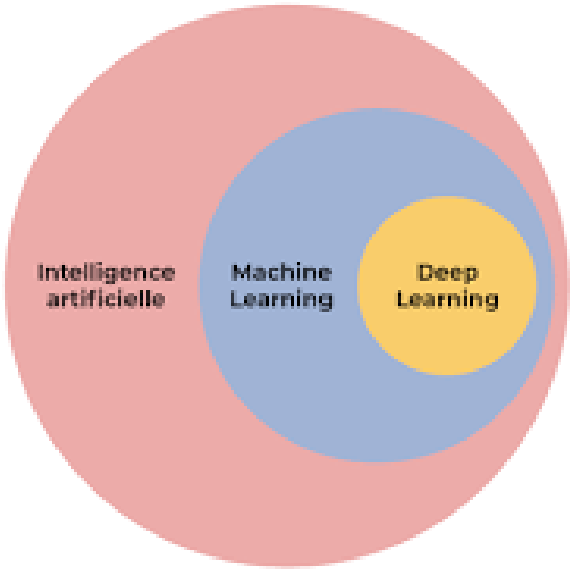
CHAPITRE 1: Revisiter les concepts clés en analyse de données

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:

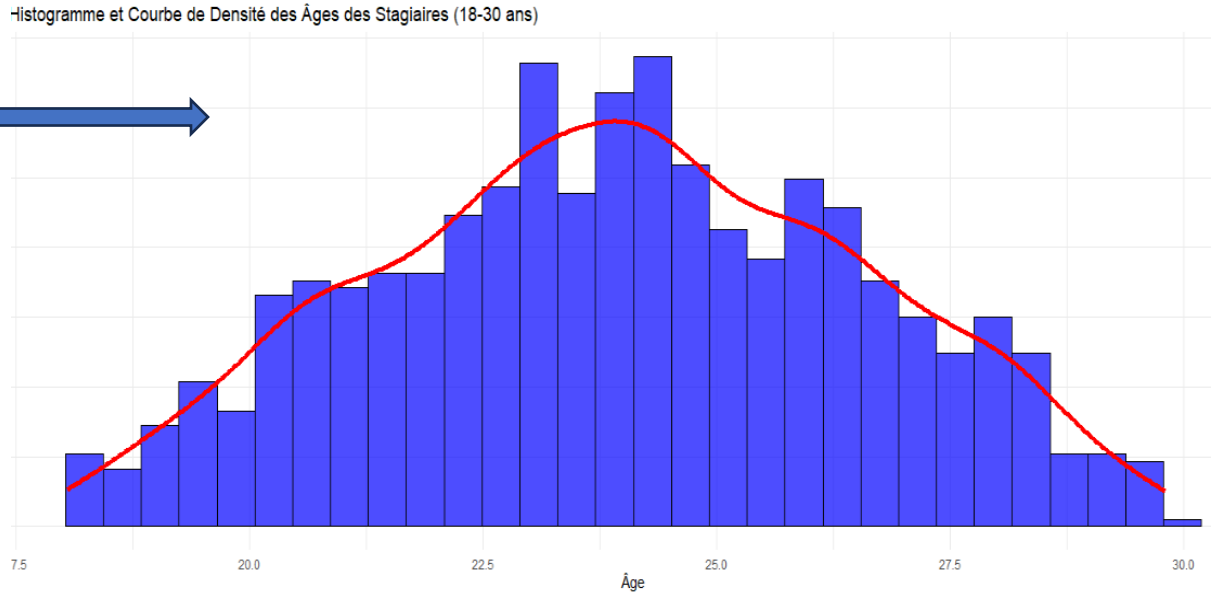
  Cas d'utilisation:

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

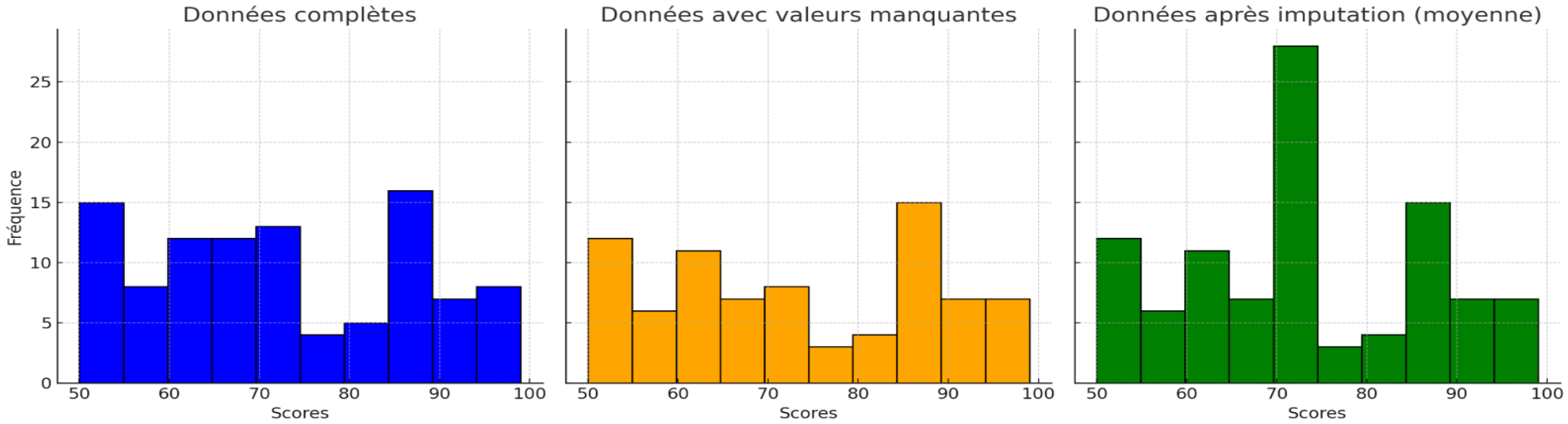
Exemple : distribution normale des âges des stagiaires.

- **Imputation avec la Moyenne:**

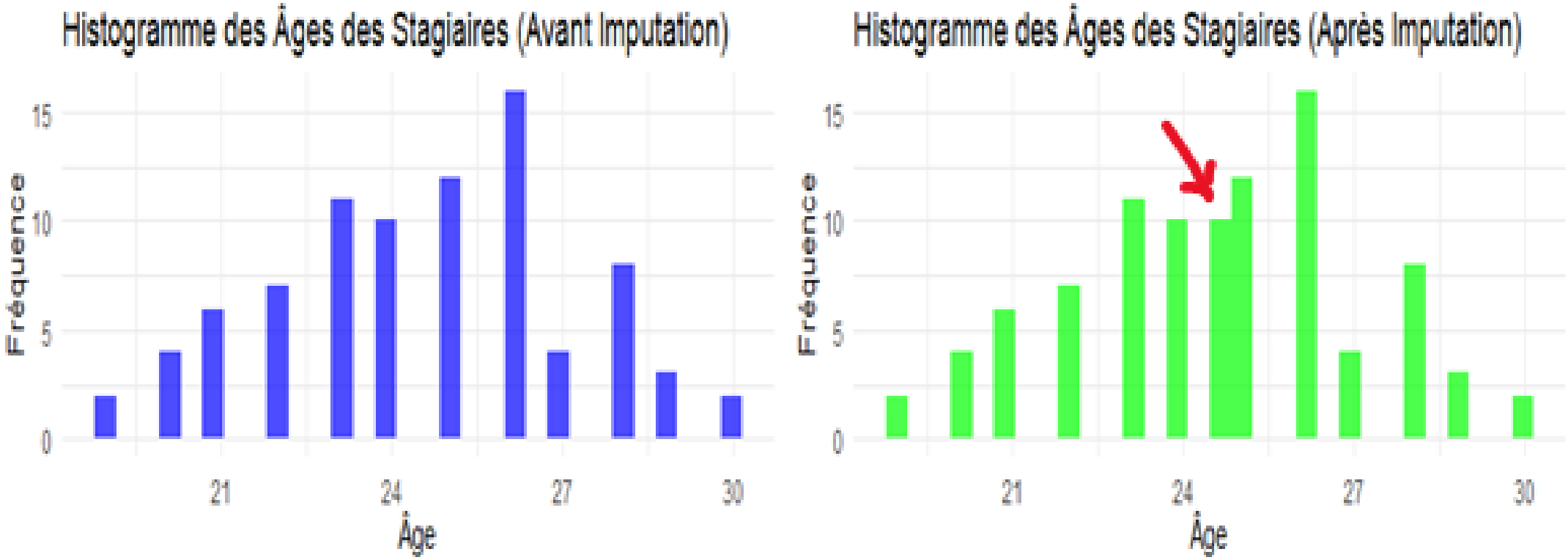
  - Distribution symétrique.



  - 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 :

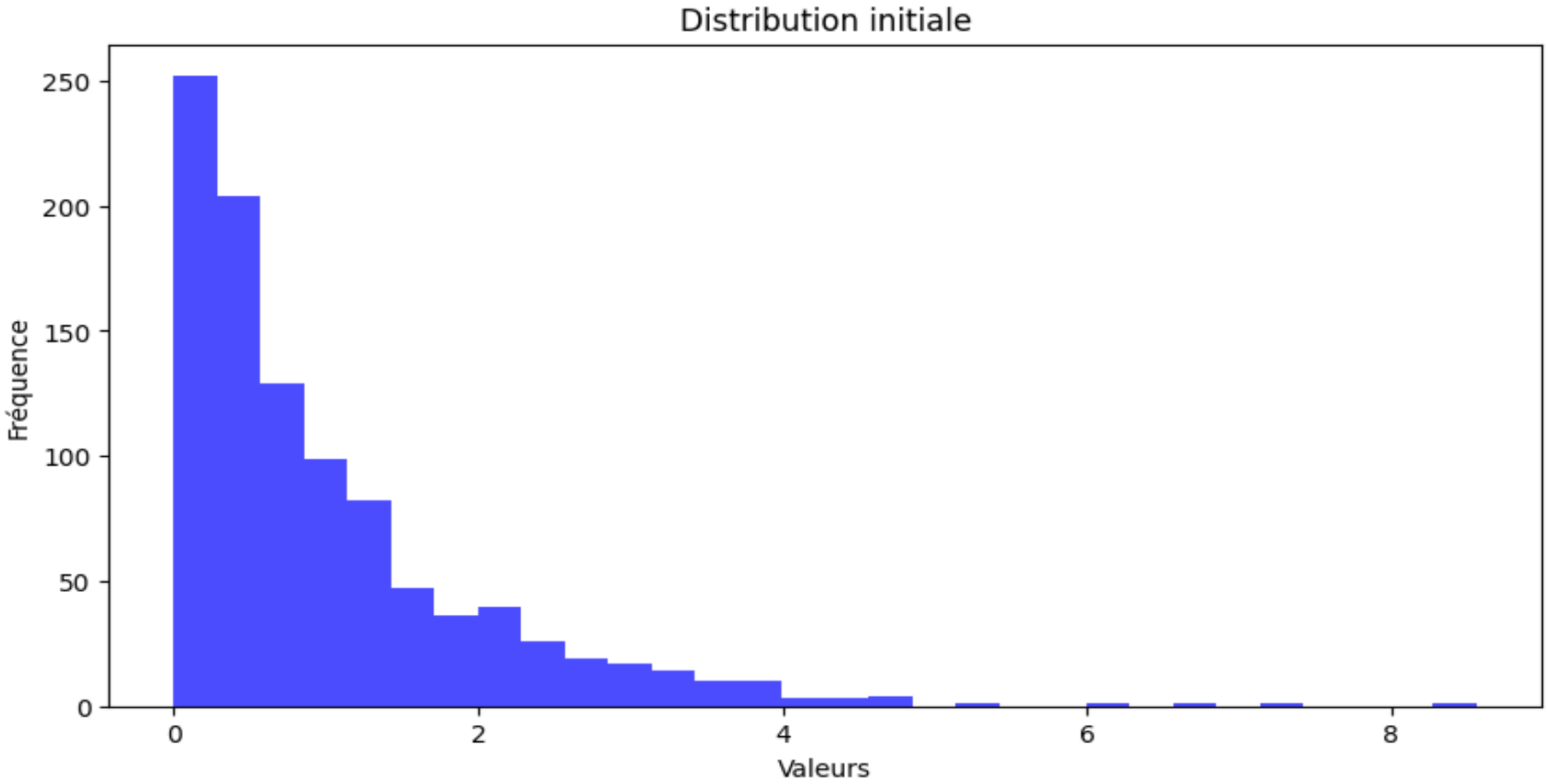
  Cas d'utilisation:

  - 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.

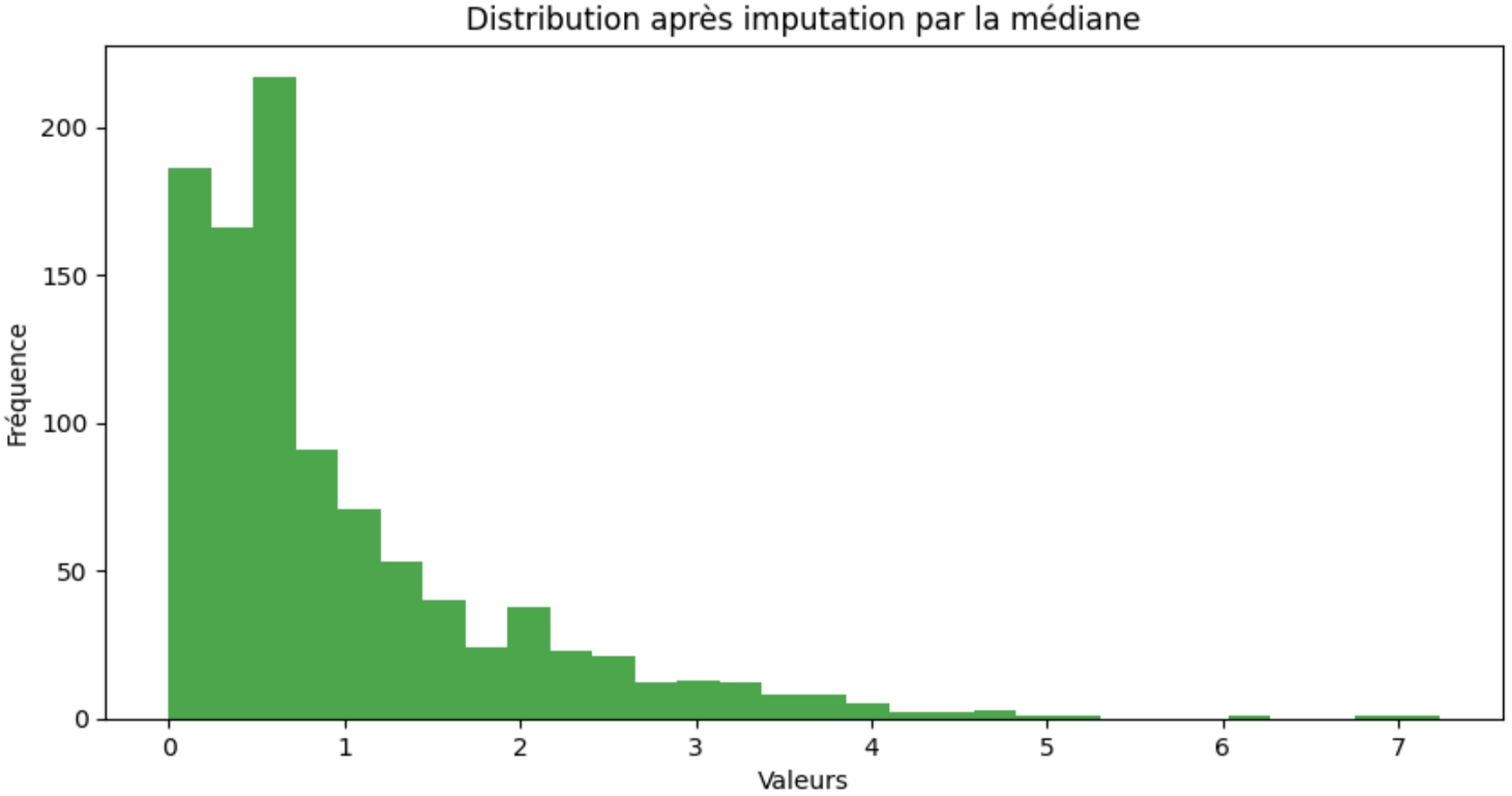
  - Nombre très faible de données manquantes < 15%. Sinon, il peut modifier la corrélation entre les variables et biaiser les modèles d’estimation.

- **Imputation avec la Médiane:**

  - Distribution asymétrique avant imputation.

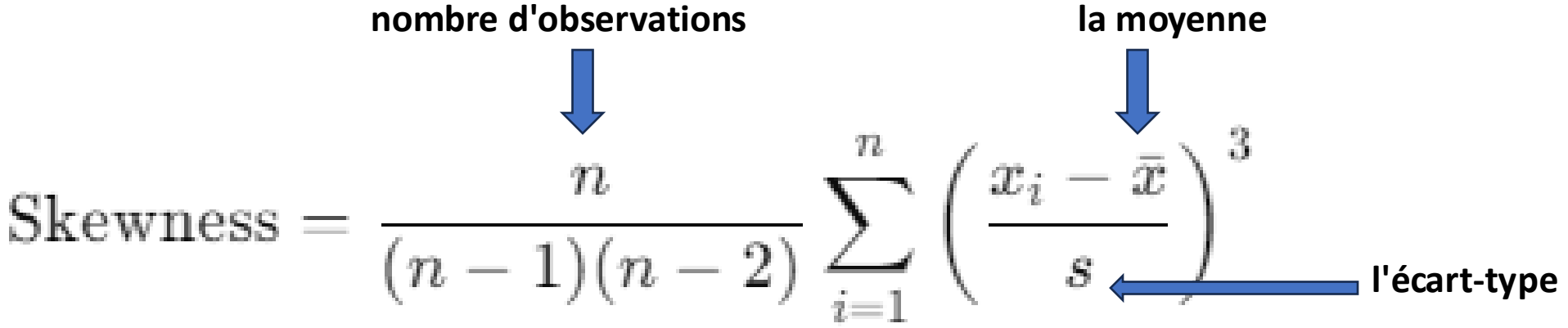


- **Imputation avec la Médiane:**
  - Distribution asymétrique après imputation.

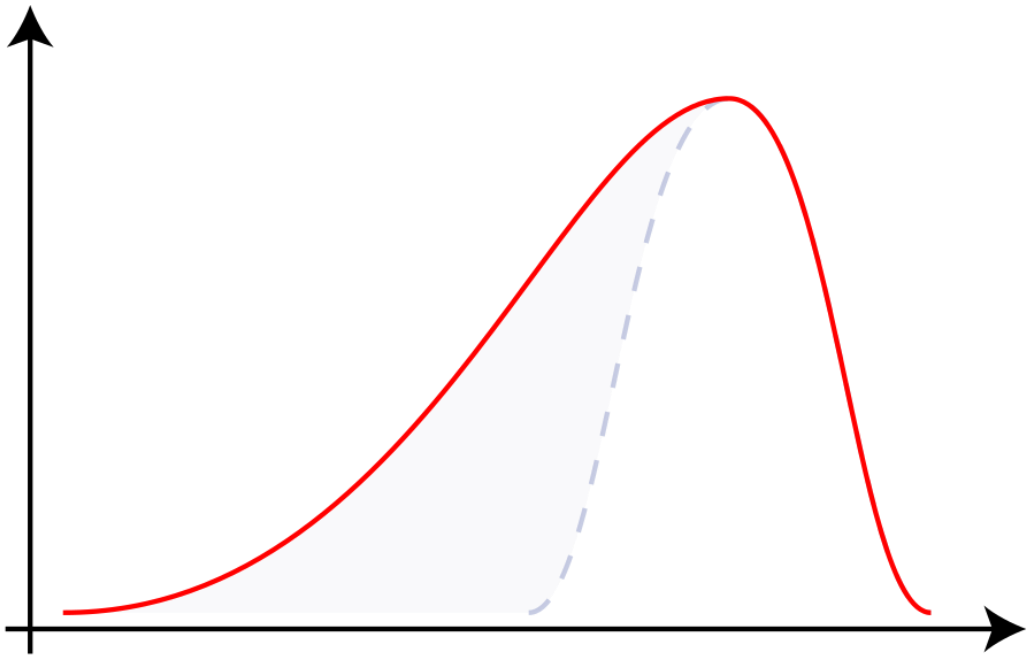


- Imputation avec la Médiane :

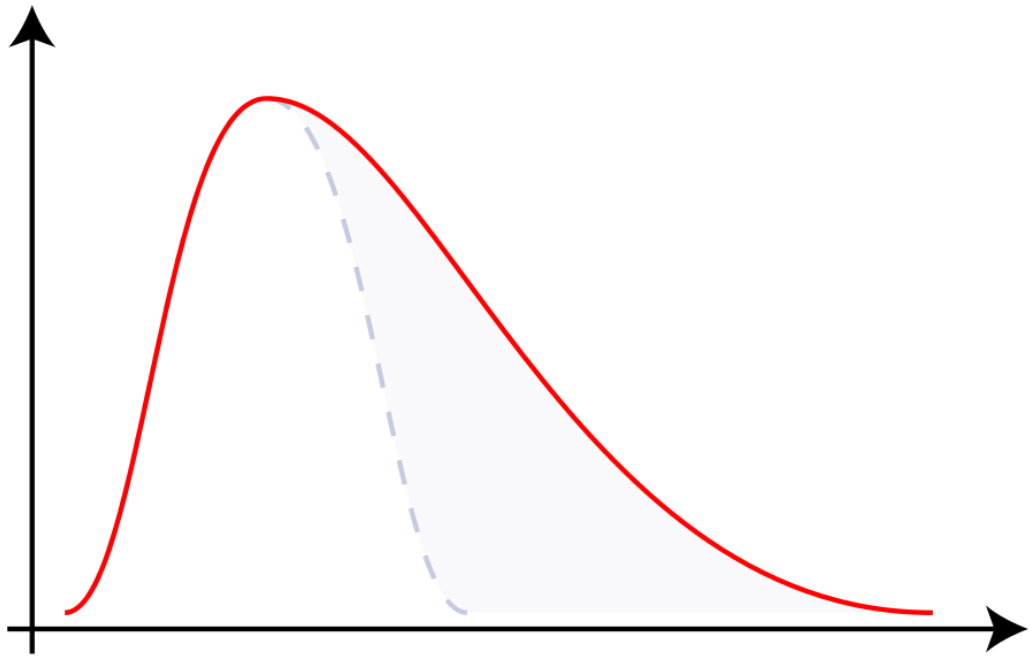
Le **skewness** d'une distribution peut être calculé avec la formule suivante



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



Negative skew

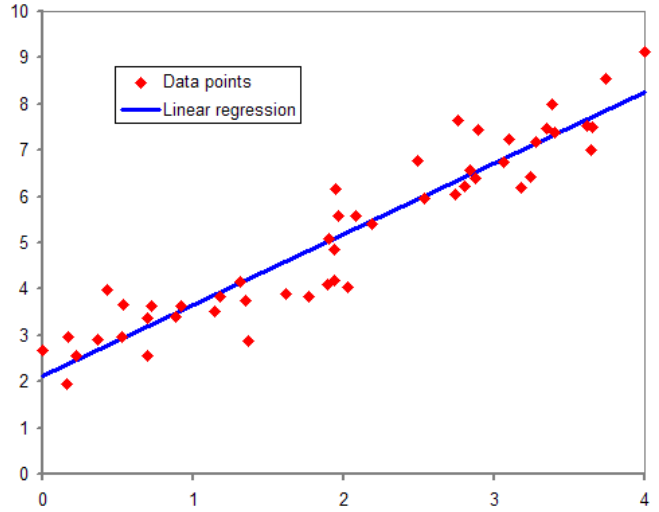


Positive skew

- **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:

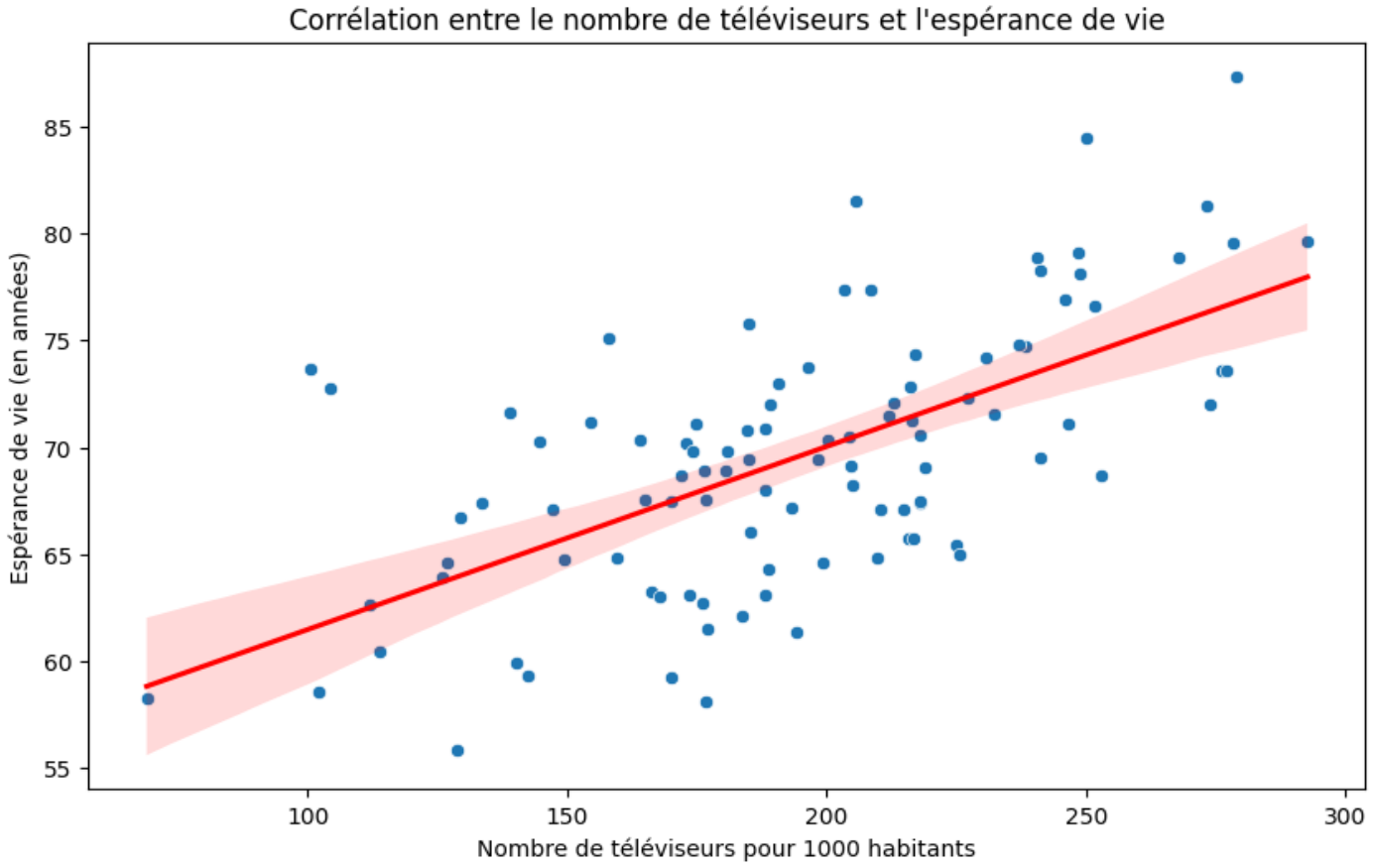
  - Existence de relations : Utilisez l'imputation par régression lorsque vous avez une ou plusieurs variables indépendantes (prédicteurs) qui sont corrélées avec la variable ayant des valeurs manquantes.

  - 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
```

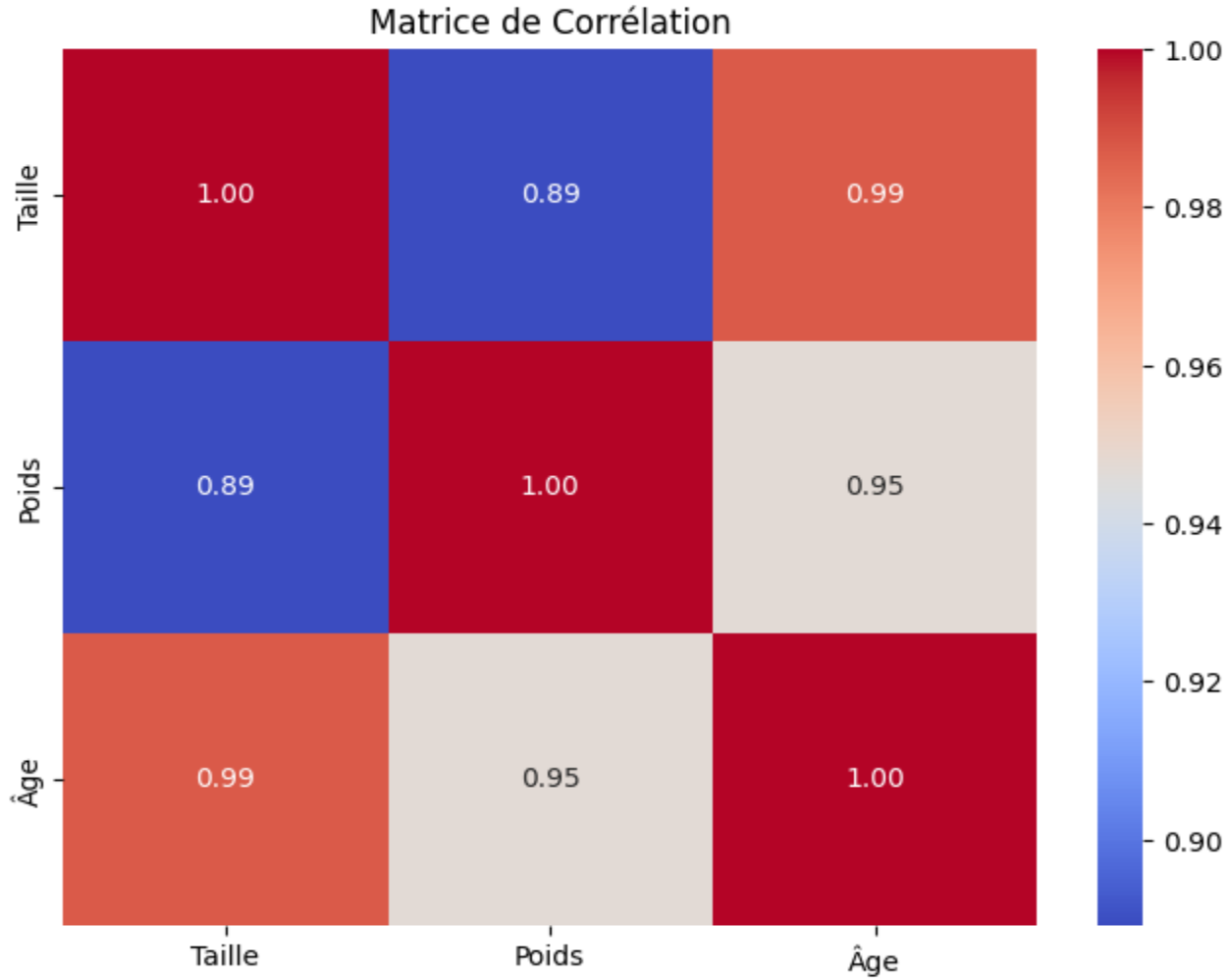
```
# représenter les données en dictionnaire
data = {
    'Taille': [170, 160, 180, 175, 185,161],
    'Poids': [65, 58, None, 75, None,64],
    'Âge': [30, 25, 40, 35, 45, 26]
}
```

```
# 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

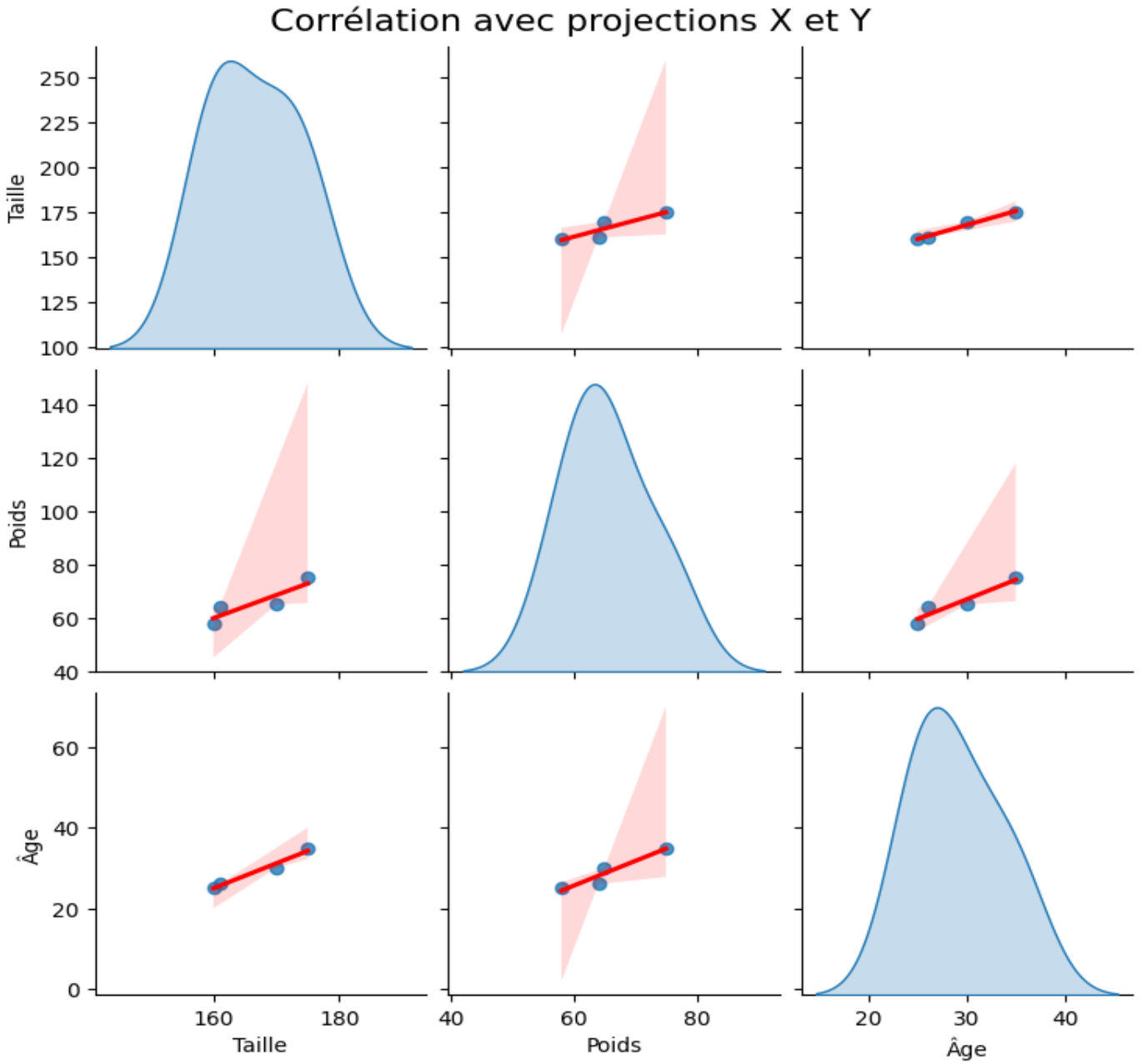


Taille (cm)	Poids(kg)	Age
170	65	30
160	58	25
180	**NaN**	40
175	75	35
185	**NaN**	45
161	**64**	26

- 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

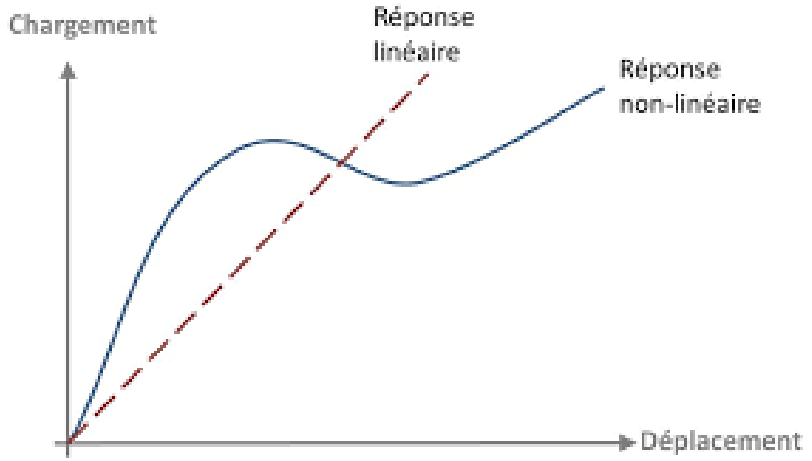
	Taille	Poids	Âge
0	170	65.000000	30
1	160	58.000000	25
2	180	86.153846	40
3	175	75.000000	35
4	185	96.923077	45
5	161	64.000000	26

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:



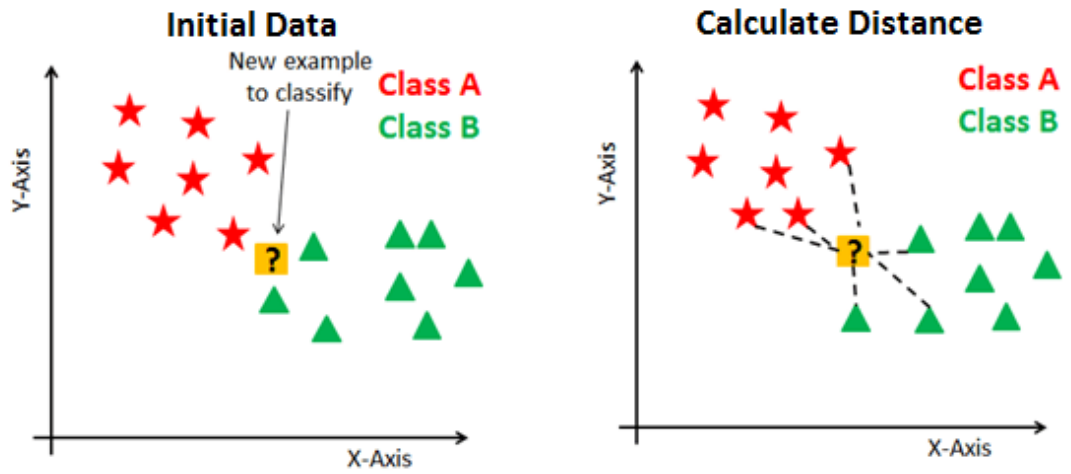
- Imputation des valeurs manquantes :

  - Imputation avec K-Nearest Neighbors (KNN) :

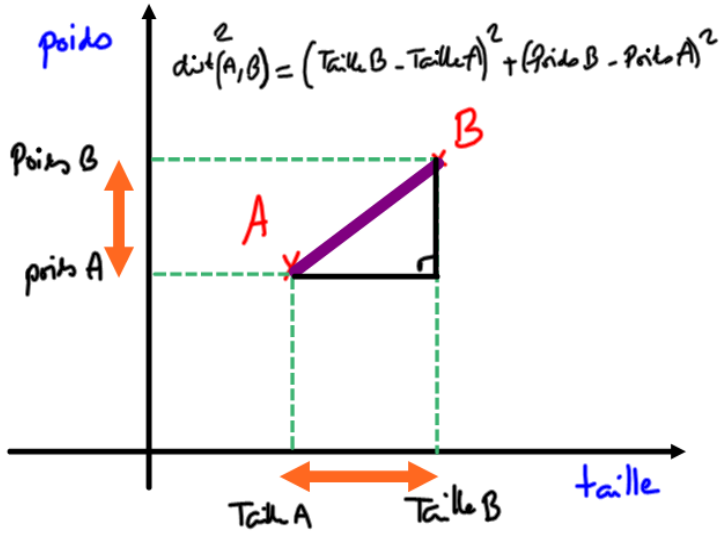
  Cas d'utilisation:

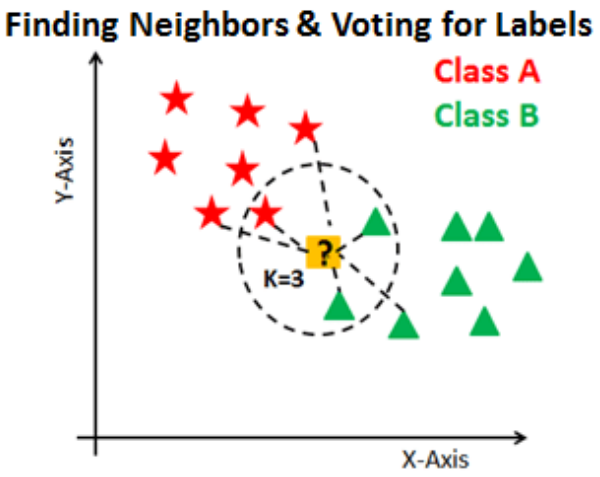
  - Il y a des relations complexes entre les variables : Si vous avez des variables qui sont liées de manière non linéaire ou complexe, KNN peut capturer ces relations sans avoir besoin de modélisation explicite.
  - Le dataset n'est pas trop grand : KNN peut être coûteux en termes de calcul, surtout pour 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



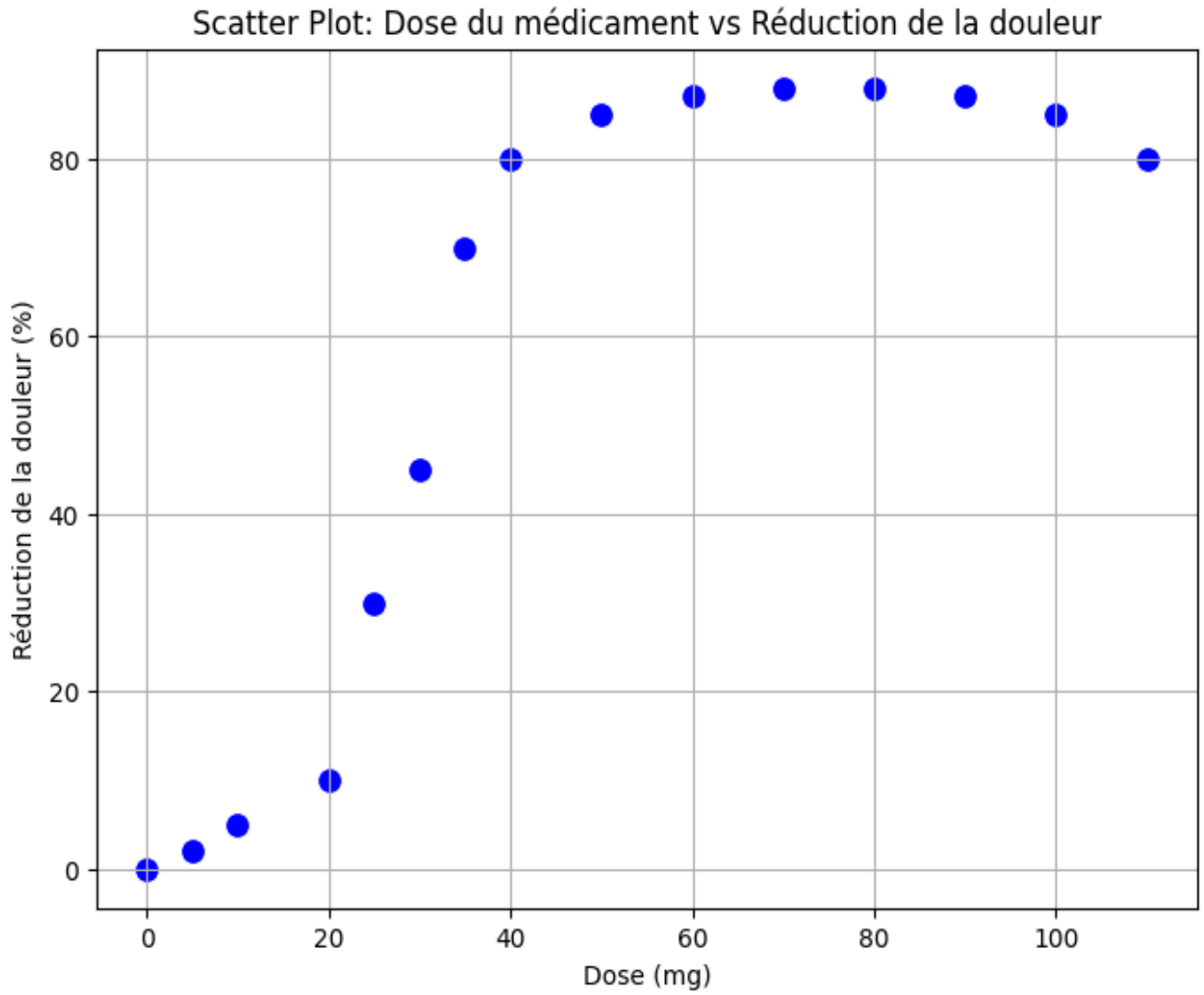


- Imputation avec KNN:

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

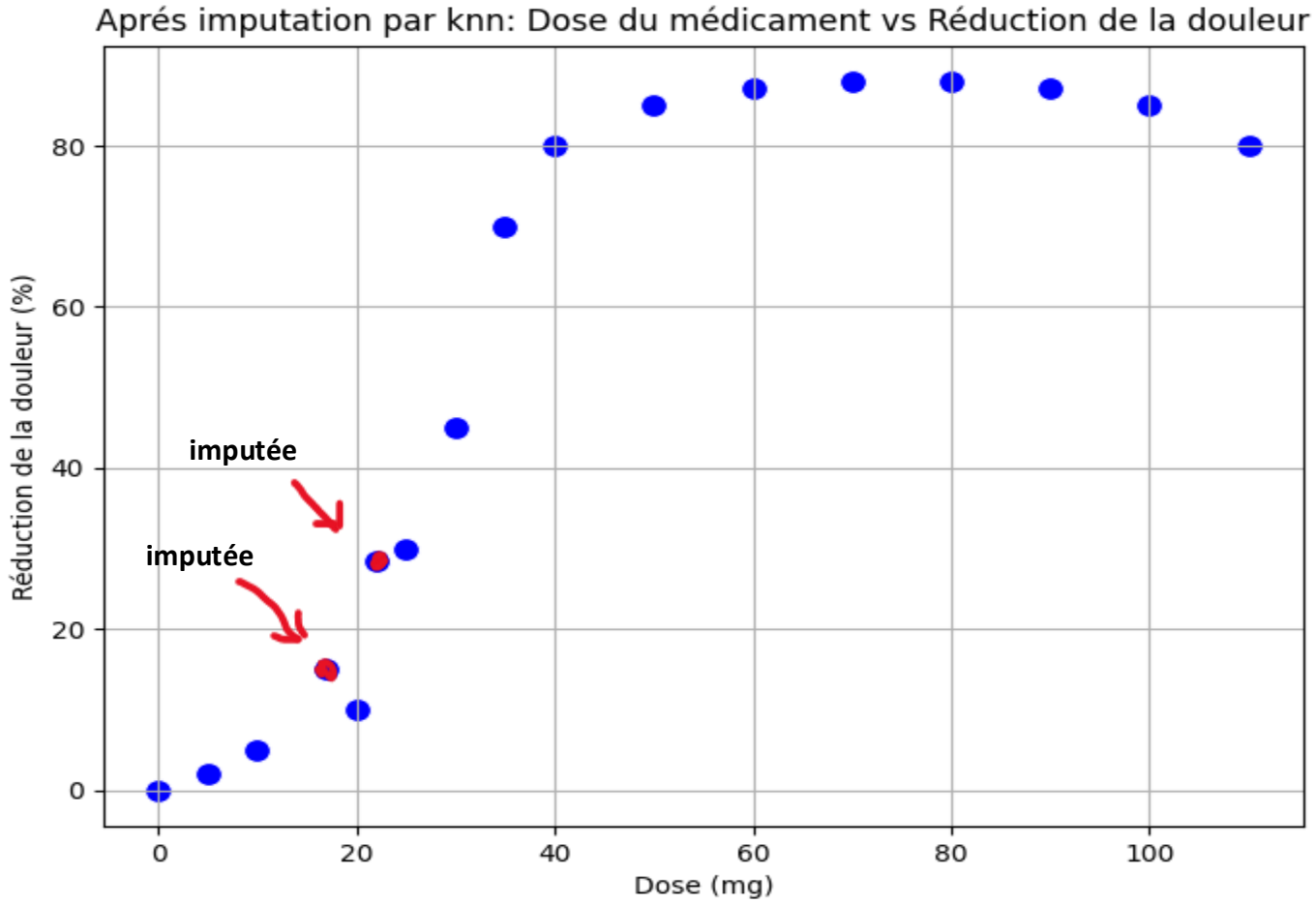


- **Exemple d'application:**:

- 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:

- 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']])
```

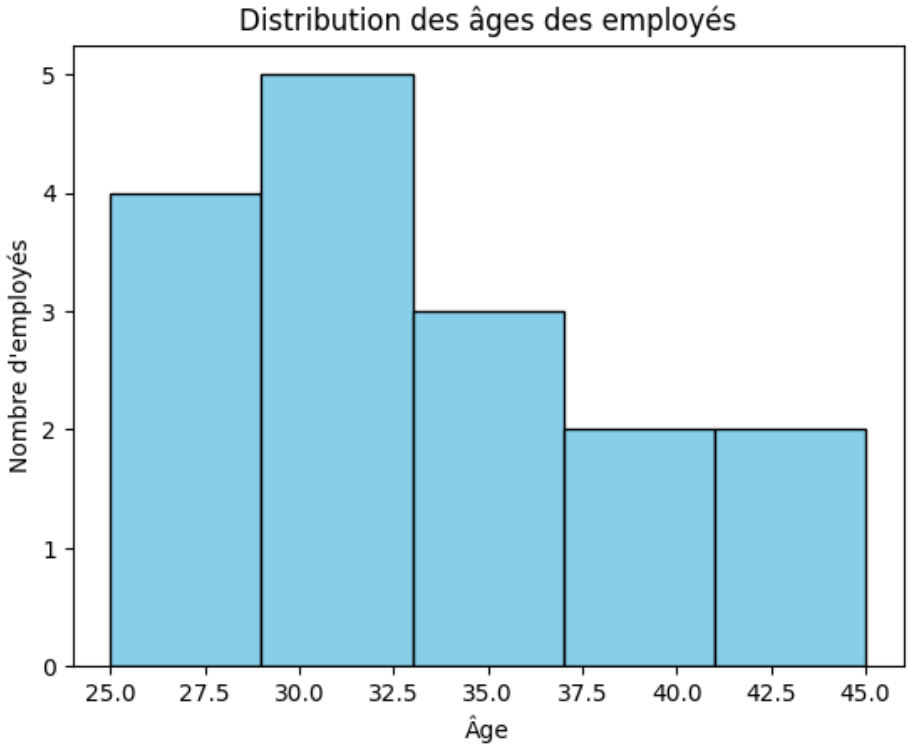
1. PRÉREQUIS de la compétence
2. Méthodes de nettoyage des données
    - Traitement des valeurs manquantes
    - Normalisation et standardisation
    - Codage des variables catégorielles
3. Analyse exploratoire des données
    - Analyse univariée
    - Analyse bivariée et corrélation
    - Techniques de visualisation de données

Techniques de visualisation de données:

1. **Histogrammes**: Représenter la distribution d'une variable continue

- Exemple : Un histogramme montrant la distribution des âges dans une entreprise permet de voir si la population est jeune, âgée ou bien répartie.

```
ages = [25, 32, 29, 30, 45, 33, 28, 35, 26, 40, 38, 32, 29, 36, 27, 41]
plt.hist(ages, bins=5, color='skyblue', edgecolor='black')
plt.xlabel('Âge')
plt.ylabel('Nombre d\'employés')
plt.title('Distribution des âges des employés')
plt.show()
```

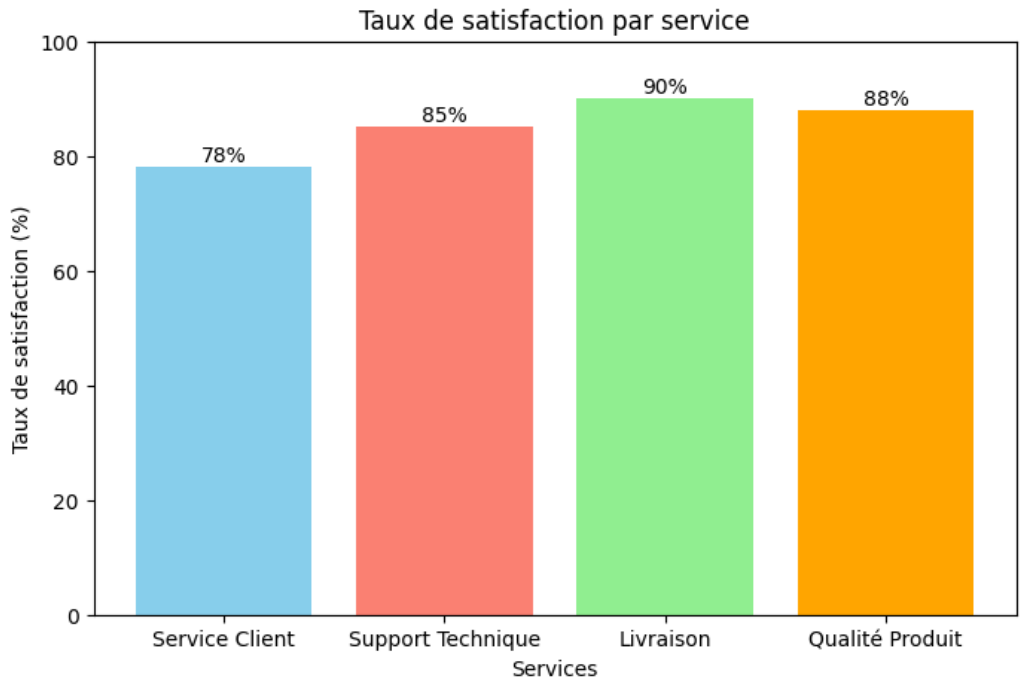


Techniques de visualisation de données:

1. Graphiques en barres: une comparaison entre différents groupes

- Exemple : Taux de satisfaction de différents services d'une entreprise

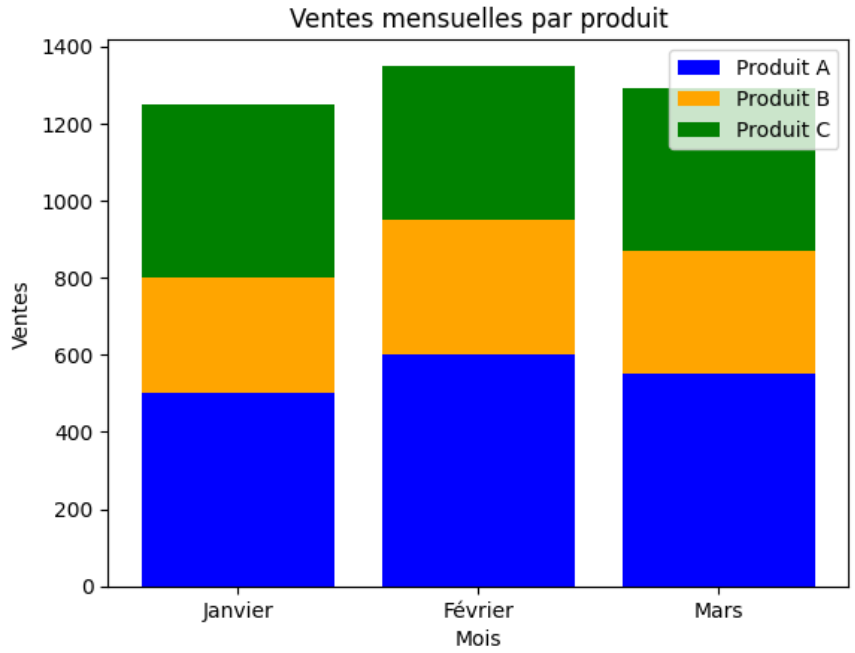
```
services = ['Service Client', 'Support Technique', 'Livraison', 'Qualité Produit']
satisfaction_scores = [78, 85, 90, 88]

# Création du graphique en barres
plt.figure(figsize=(8, 5))
plt.bar(services, satisfaction_scores, color=['skyblue', 'salmon', 'lightgreen', 'orange'])
plt.xlabel('Services')
plt.ylabel('Taux de satisfaction (%)')
plt.title('Taux de satisfaction par service')
plt.ylim(0, 100)

# Affichage des valeurs sur chaque barre
for i, score in enumerate(satisfaction_scores):
    plt.text(i, score + 1, f"{score}%", ha='center')
plt.show()
```



Techniques de visualisation de données:

1. Graphiques en barres: une comparaison entre différents groupes
   - Exemple : Comparer les ventes mensuelles par produit pour identifier le produit le plus populaire.

```
months = ['Janvier', 'Février', 'Mars']
product_a = [500, 600, 550]
product_b = [300, 350, 320]
product_c = [450, 400, 420]
plt.bar(months, product_a, label='Produit A', color='blue')
plt.bar(months, product_b, bottom=product_a, label='Produit B', color='orange')
plt.bar(months, product_c, bottom=np.array(product_a)+np.array(product_b), label='Produit C', color='green')
plt.xlabel('Mois')
plt.ylabel('Ventes')
plt.title('Ventes mensuelles par produit')
plt.legend()
plt.show
```

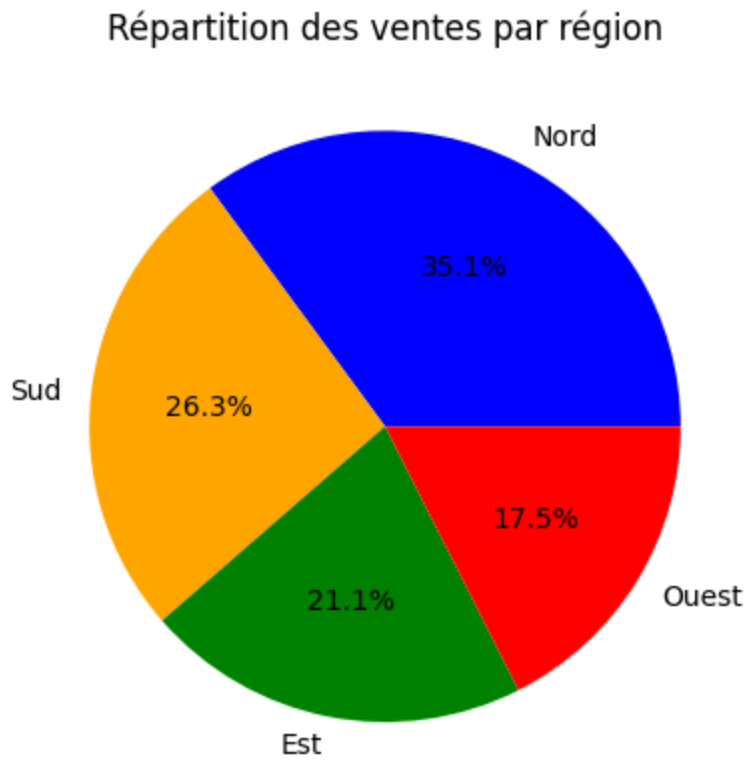


Techniques de visualisation de données:

1. Graphiques en secteurs: Montrer la proportion des différentes catégories dans un ensemble

- Exemple : la répartition du chiffre d'affaires par région

```
regions = ['Nord', 'Sud', 'Est', 'Ouest']
sales = [200000, 150000, 120000, 100000]

plt.pie(sales, labels=regions, autopct='%1.1f%%', colors=['blue', 'orange', 'green', 'red'])

plt.title('Répartition des ventes par région')
plt.show()
```

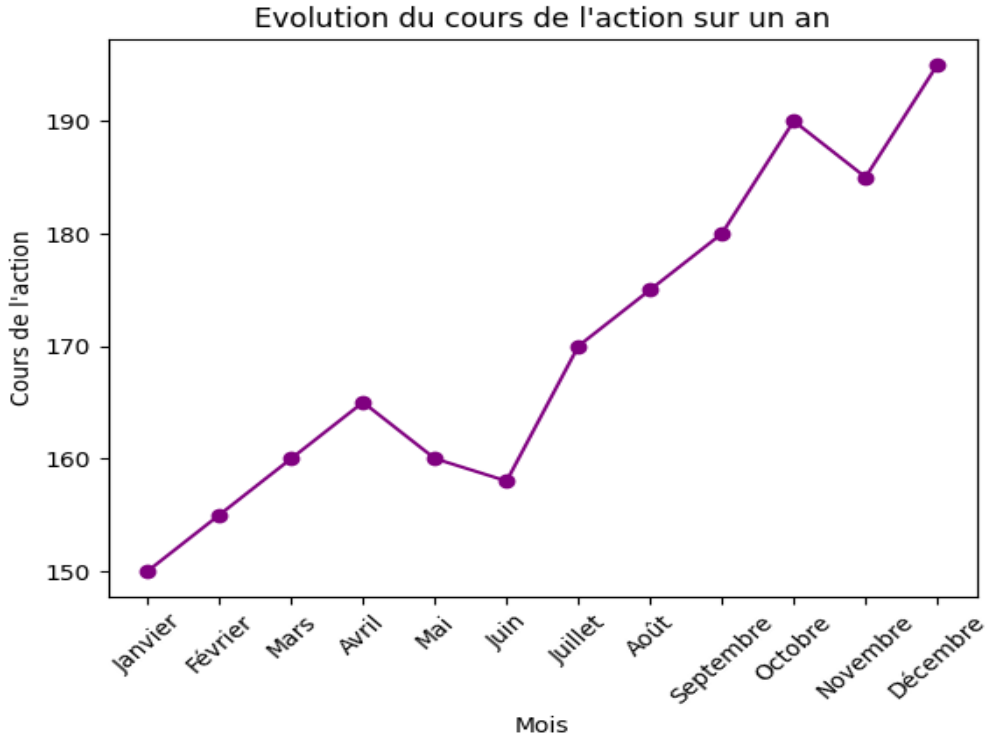


Techniques de visualisation de données:

1. Diagramme en ligne (Line plot): Visualiser les tendances temporelles ou les évolutions dans le temps.

- Exemple : Suivre l'évolution des actions d'une entreprise sur plusieurs années.

```
months = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Août', 'Septembre', 'Octobre', 'Novembre', 'Décembre']
stock_prices = [150, 155, 160, 165, 160, 158, 170, 175, 180, 190, 185, 195]

plt.plot(months, stock_prices, marker='o', color='purple')

plt.xlabel('Mois')
plt.ylabel('Cours de l\'action')
plt.title('Evolution du cours de l\'action sur un an')
plt.xticks(rotation=45)
plt.show()
```

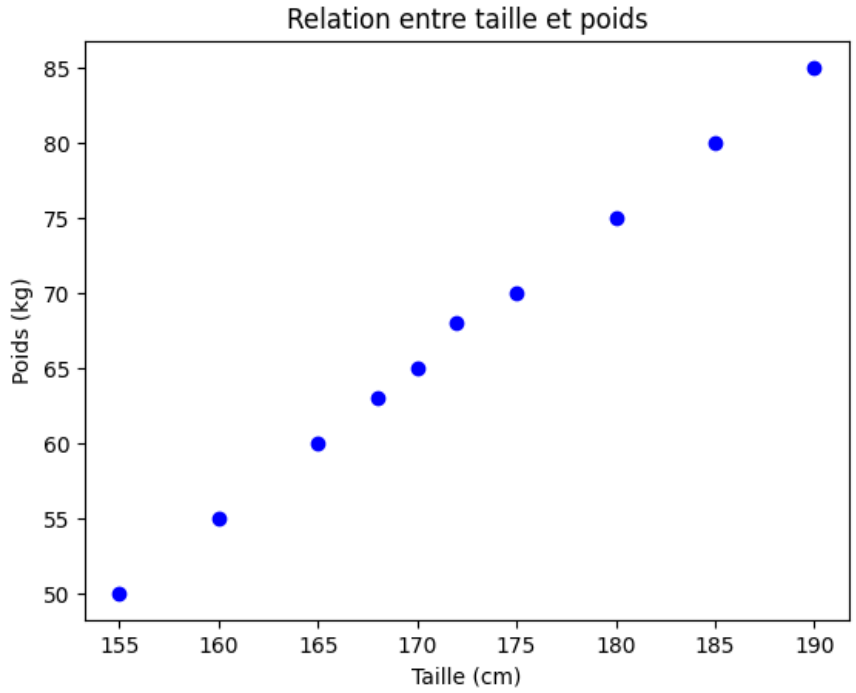


Techniques de visualisation de données:

1. Nuages de points (Scatter plots): Analyser la relation entre deux variables continues

- Exemple : Comparer la taille et le poids d'un groupe d'individus.

```
height = [160, 170, 180, 175, 165, 155, 168, 172, 185, 190]
weight = [55, 65, 75, 70, 60, 50, 63, 68, 80, 85]

plt.scatter(height, weight, color='blue')

plt.xlabel('Taille (cm)')
plt.ylabel('Poids (kg)')
plt.title('Relation entre taille et poids')
plt.show()
```

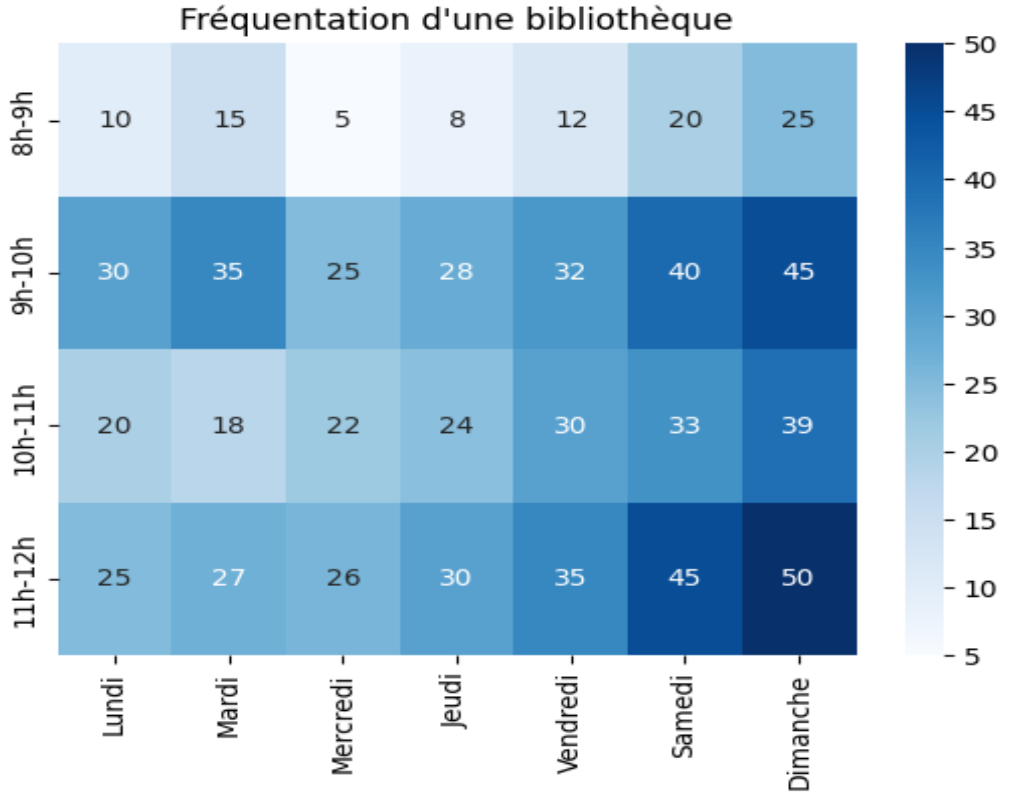


Techniques de visualisation de données:

1. Carte de chaleur (Heatmap): Montrer des valeurs dans une matrice de données à l'aide de couleurs

- Exemple : Fréquentation d'une bibliothèque.

```
data = np.array([[10, 15, 5, 8, 12, 20, 25],
                 [30, 35, 25, 28, 32, 40, 45],
                 [20, 18, 22, 24, 30, 33, 39],
                 [25, 27, 26, 30, 35, 45, 50]])
days = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi', 'Dimanche']
hours = ['8h-9h', '9h-10h', '10h-11h', '11h-12h']

sns.heatmap(data, annot=True, xticklabels=days, yticklabels=hours, cmap='Blues')

plt.title('Fréquentation d\'une bibliothèque')
plt.show()
```

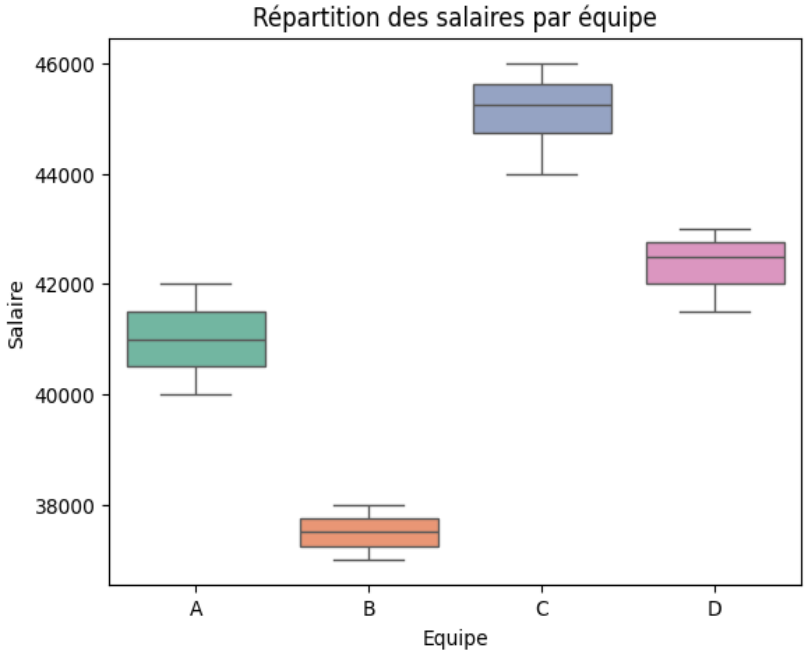


Techniques de visualisation de données:

1. Graphique en boîte (Box plot): Résumer la distribution d'une variable avec des informations sur la médiane, les quartiles et les valeurs extrêmes.

- Exemple : Comparer la répartition des salaires dans différentes équipes d'une entreprise pour repérer les écarts.

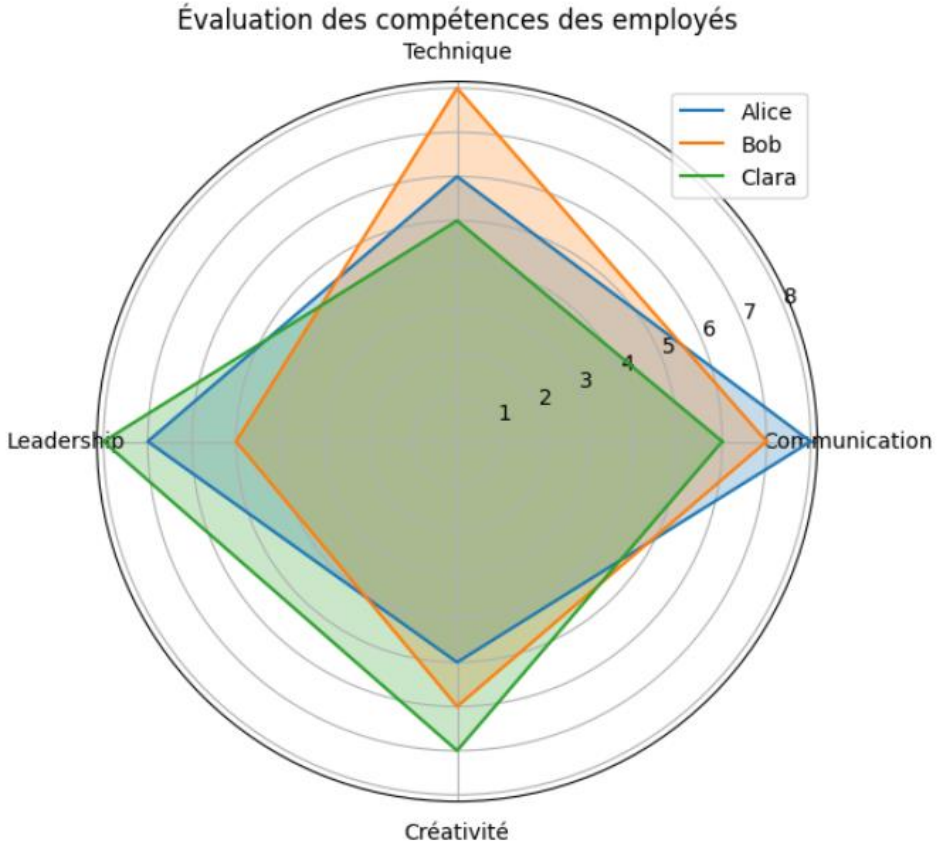
```
data = { 'Equipe': ['A', 'A', 'A', 'B', 'B', 'C', 'C', 'C', 'C', 'D', 'D', 'D'],
         'Salaire': [40000, 42000, 41000, 38000, 37000, 45000, 44000, 46000, 45500, 43000, 42500, 41500] }
df = pd.DataFrame(data)

sns.boxplot(x='Equipe', y='Salaire', data=df, palette='Set2')

plt.title('Répartition des salaires par équipe')
plt.show()
```



Techniques de visualisation de données:

1. Diagrammes en radar: Comparer plusieurs variables pour différents individus ou groupes sur un graphique à plusieurs axes
   - Exemple : Évaluer les compétences d'employés en fonction de plusieurs critères.

```
from math import pi
data = { 'Metrics': ['Communication', 'Technique', 'Leadership', 'Créativité'],
         'Alice': [8, 6, 7, 5],
         'Bob': [7, 8, 5, 6],
         'Clara': [6, 5, 8, 7] }
df = pd.DataFrame(data)
categories = list(df['Metrics'])
N = len(categories)
angles = [n / float(N) * 2 * pi for n in range(N)]
angles += angles[:1]
fig, ax = plt.subplots(figsize=(6, 6), subplot_kw=dict(polar=True))
for name, values in df.drop('Metrics', axis=1).items():
    values = values.tolist()
    values += values[:1]
    ax.plot(angles, values, label=name)
   ax.fill(angles, values, alpha=0.25)
   ax.set_xticks(angles[:-1])
   ax.set_xticklabels(categories)
plt.title('Évaluation des compétences des employés')
plt.legend()
plt.show()
```



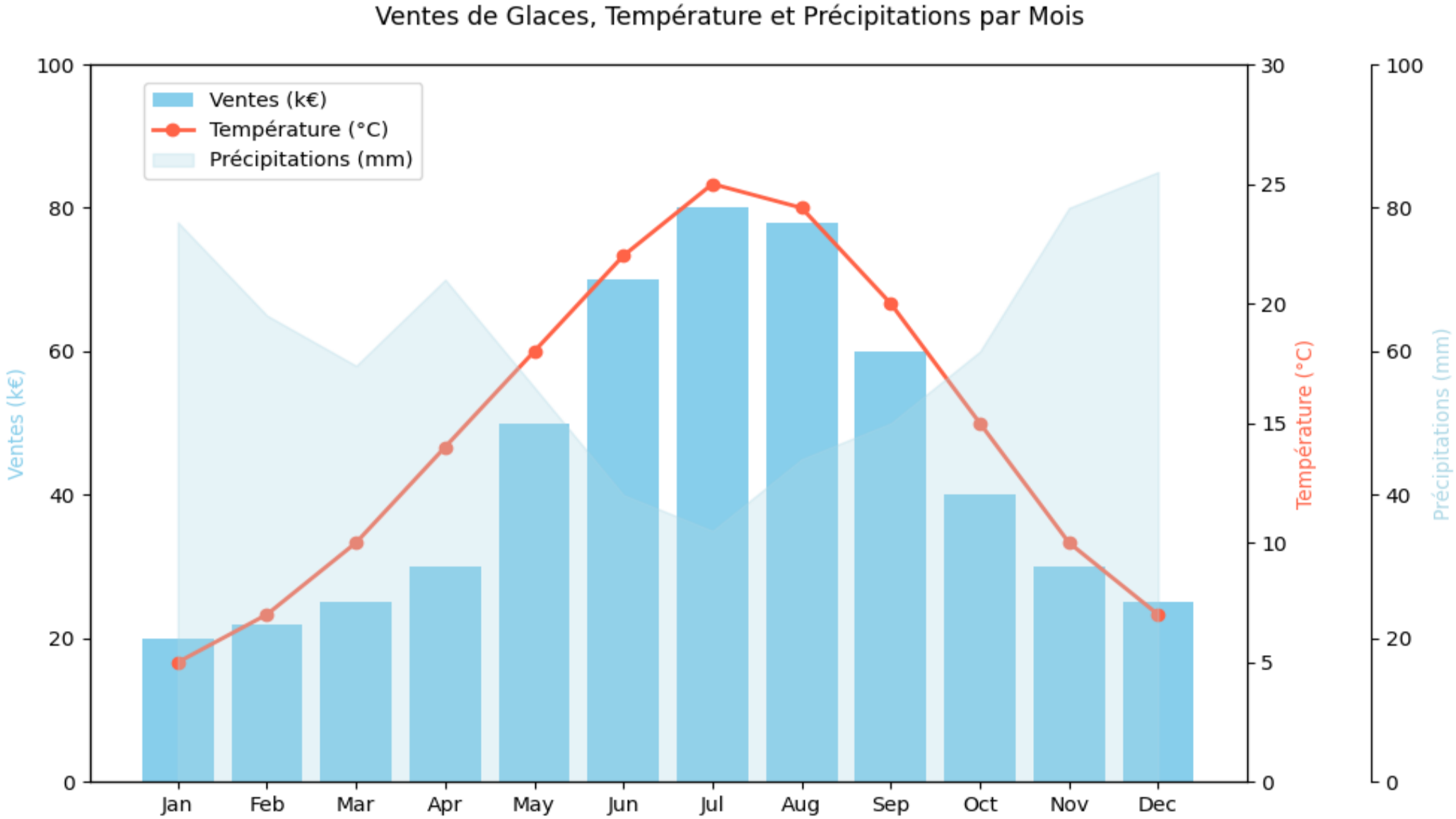
1. PRÉREQUIS de la compétence
2. Méthodes de nettoyage des données
    - Traitement des valeurs manquantes
    - Normalisation et standardisation
    - Codage des variables catégorielles
3. Analyse exploratoire des données
    - Analyse univariée
    - Analyse bivariée et corrélation
4. Techniques de visualisation de données
    - Maîtriser la visualisation avancée des données
    - Graphiques 3D et interactifs
    - Visualisations complexes

Maîtriser la visualisation avancée des données:

1. **Graphiques multi-axes et combinés** : Utile pour comparer plusieurs séries de données ayant des échelles différentes.

- Exemple : les ventes mensuelles de glaces et la météo.

	Mois	Ventes (k€)	Température (°C)	Précipitations (mm)
0	Jan	20	5	78
1	Feb	22	7	65
2	Mar	25	10	58
3	Apr	30	14	70
4	May	50	18	55
5	Jun	70	22	40
6	Jul	80	25	35
7	Aug	78	24	45
8	Sep	60	20	50
9	Oct	40	15	60
10	Nov	30	10	80
11	Dec	25	7	85


Ventes de Glaces, Température et Précipitations par Mois
Ventes (k€)
Température (°C)
Précipitations (mm)
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
0
20
40
60
80
100
0
5
10
15
20
25
30
0
20
40
60
80
100

```
import matplotlib.pyplot as plt
import pandas as pd
# Jeu de données
data = {
'Mois': ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'], 'Ventes (k€)': [20, 22, 25, 30, 50, 70, 80, 78, 60, 40, 30, 25],
'Température (°C)': [5, 7, 10, 14, 18, 22, 25, 24, 20, 15, 10, 7],
'Précipitations (mm)': [78, 65, 58, 70, 55, 40, 35, 45, 50, 60, 80, 85]
}
df = pd.DataFrame(data)
# Configuration du graphique
fig, ax1 = plt.subplots(figsize=(10, 6))
# Graphique à barres pour les ventes
ax1.bar(df['Mois'], df['Ventes (k€)'], color='skyblue', label='Ventes (k€)')
ax1.set_ylabel("Ventes (k€)", color='skyblue')
ax1.set_ylim(0, 100)
```

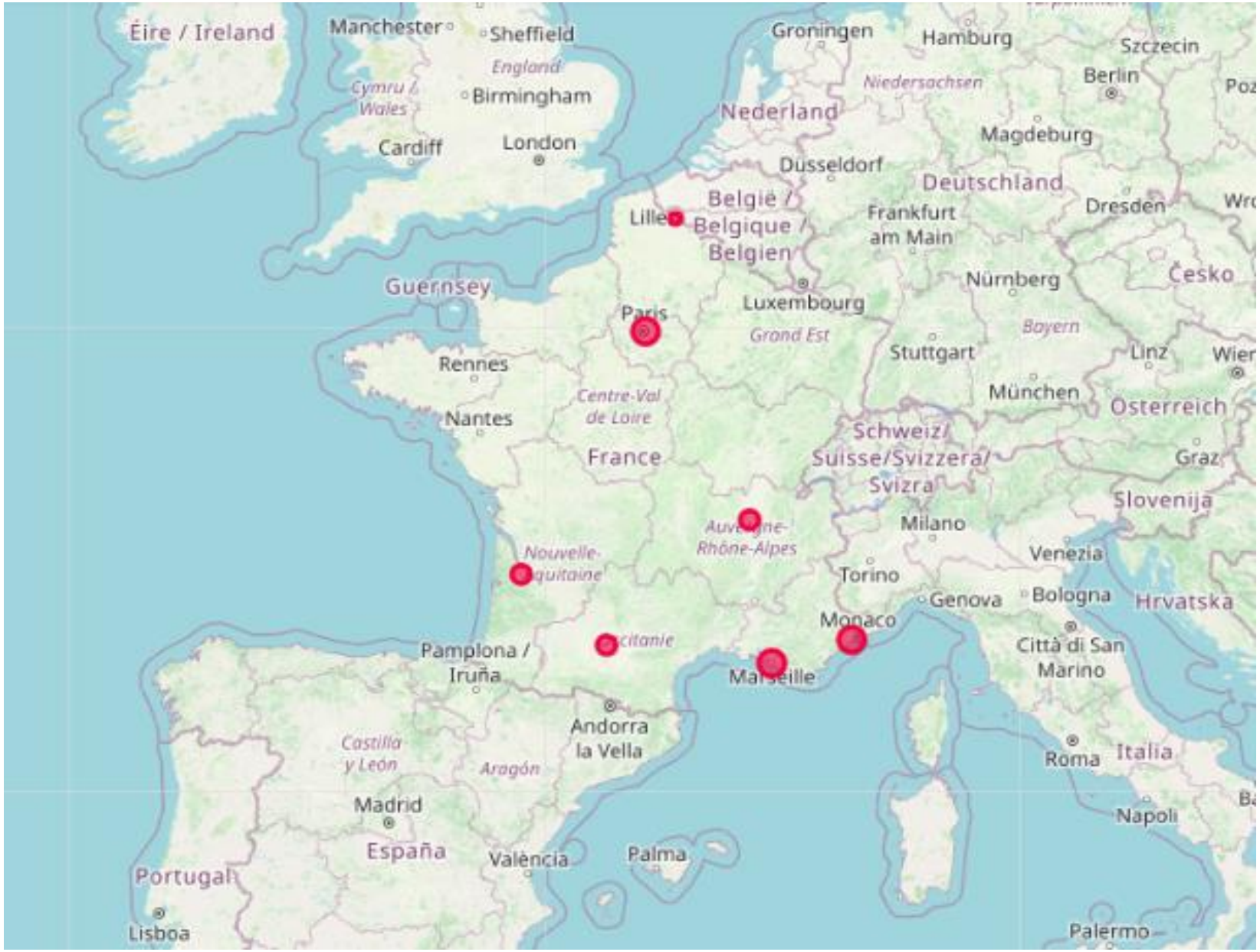
```
# Deuxième axe pour la température
ax2 = ax1.twinx()
ax2.plot(df['Mois'], df['Température (°C)'], color='tomato', marker='o', linestyle='-', linewidth=2, label='Température (°C)')
ax2.set_ylabel("Température (°C)", color='tomato')
ax2.set_ylim(0, 30)
# Graphique en aires pour les précipitations
ax3 = ax1.twinx()
ax3.fill_between(df['Mois'], df['Précipitations (mm)'], color='lightblue', alpha=0.3, label='Précipitations (mm)')
ax3.spines['right'].set_position(('outward', 60)) # Décale l'axe des précipitations
ax3.set_ylabel("Précipitations (mm)", color='lightblue')
ax3.set_ylim(0, 100)
# Ajouter des légendes et un titre
fig.suptitle("Ventes de Glaces, Température et Précipitations par Mois")
fig.tight_layout()
fig.legend(loc="upper left", bbox_to_anchor=(0.1, 0.9))
plt.show()
```

Maîtriser la visualisation avancée des données:

2. **Cartographie et géovisualisation**: Utile pour des analyses géospatiales, en utilisant des bibliothèques comme Folium ou GeoPandas.

- Exemple : Taux de Criminalité par Ville.

Ville	Latitude	Longitude	Taux de Criminalité
Paris	48.8566	2.3522	650
Lyon	45.7640	4.8357	480
Marseille	43.2965	5.3698	700
Lille	50.6292	3.0573	300
Bordeaux	44.8378	-0.5792	450
Toulouse	43.6047	1.4442	520
Nice	43.7102	7.2620	680


Éire / Ireland
Manchester
Sheffield
England
Cymru / Wales
Birmingham
Cardiff
London
Groningen
Hamburg
Szczecin
Niedersachsen
Berlin
Nederland
Magdeburg
Düsseldorf
Deutschland
België / Belgique / Belgien
Lille
Frankfurt am Main
Dresden
Guernsey
Nürnberg
Česko
Luxembourg
Paris
Grand Est
Bayern
Stuttgart
Linz
Rennes
München
Centre-Val de Loire
Nantes
Österreich
Schweiz/ Suisse/Svizzera/ Svizra
France
Graz
Slovenija
Milano
Auvergne-Rhône-Alpes
Venezia
Nouvelle-Aquitaine
Torino
Genova
Bologna
Hrvatska
Monaco
Occitanie
Città di San Marino
Pamplona / Iruña
Marseille
Andorra la Vella
Roma
Italia
Castilla y León
Aragón
Napoli
Madrid
España
València
Palma
Portugal
Lisboa
Palermo

```
import folium
from folium import plugins

# Créer une carte centrée sur la France
map_france = folium.Map(location=[46.603354, 1.888334], zoom_start=5)
# Jeu de données
data = {
'Ville': ['Paris', 'Lyon', 'Marseille', 'Lille', 'Bordeaux', 'Toulouse', 'Nice'],
'Latitude': [48.8566, 45.7640, 43.2965, 50.6292, 44.8378, 43.6047, 43.7102],
'Longitude': [2.3522, 4.8357, 5.3698, 3.0573, -0.5792, 1.4442, 7.2620],
'Taux de Criminalité': [650, 480, 700, 300, 450, 520, 680] }
# Afficher la carte
 map_france
```

```
    # Couleur et taille du cercle selon le taux de criminalité
    folium.CircleMarker( location=[lat, lon],
                         radius=crime_rate / 100,
                        # Taille proportionnelle au taux de criminalité
                         color='crimson',
                        fill=True,
                        fill_color='crimson',
                        fill_opacity=0.6,
                       popup=f"{ville}: {crime_rate} crimes"
                      ).add_to(map_france)
# Afficher la carte
 map_france
```

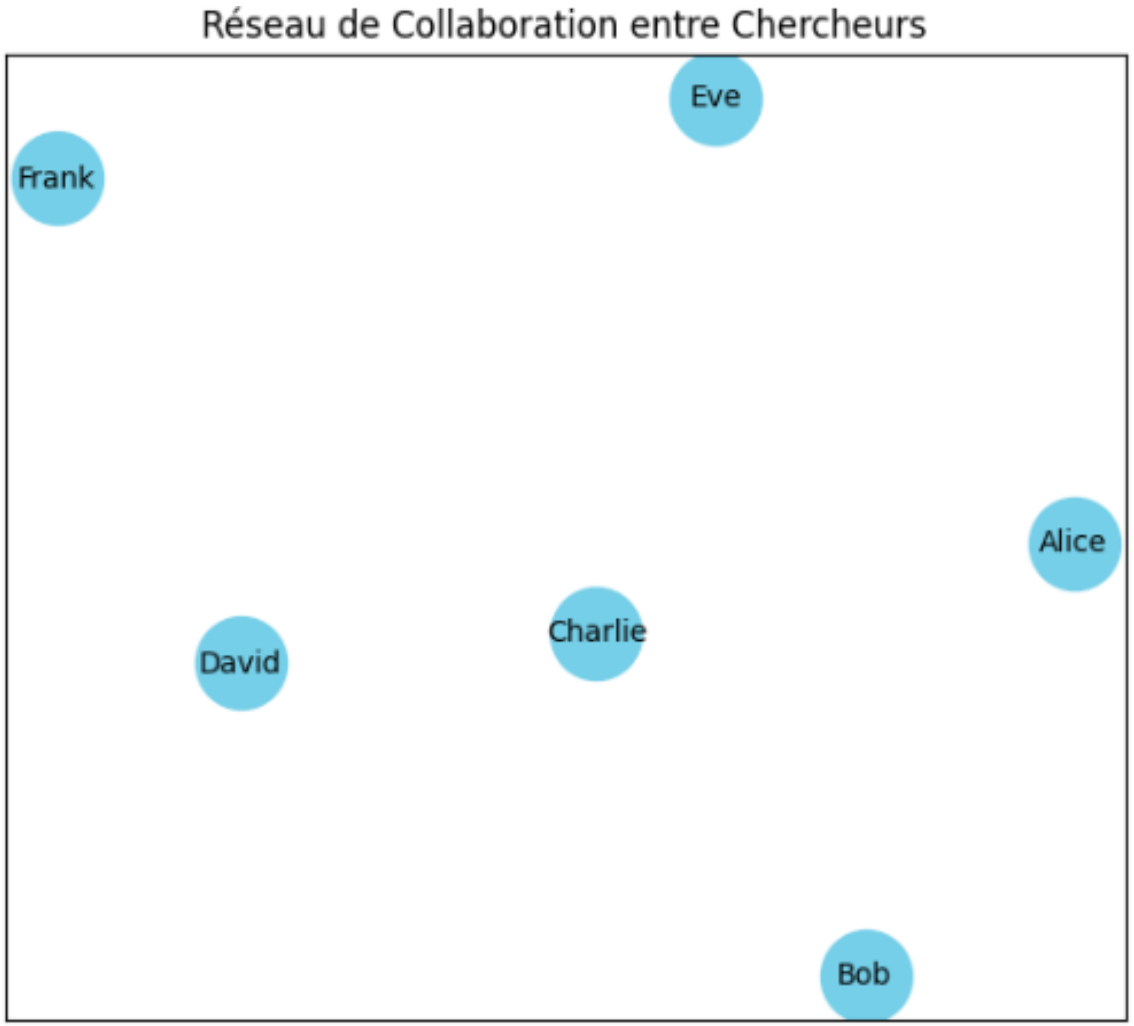
Maîtriser la visualisation avancée des données:

3. Diagrammes en réseau: Utile pour visualiser les relations complexes entre les entités, par exemple pour des données de réseau social.

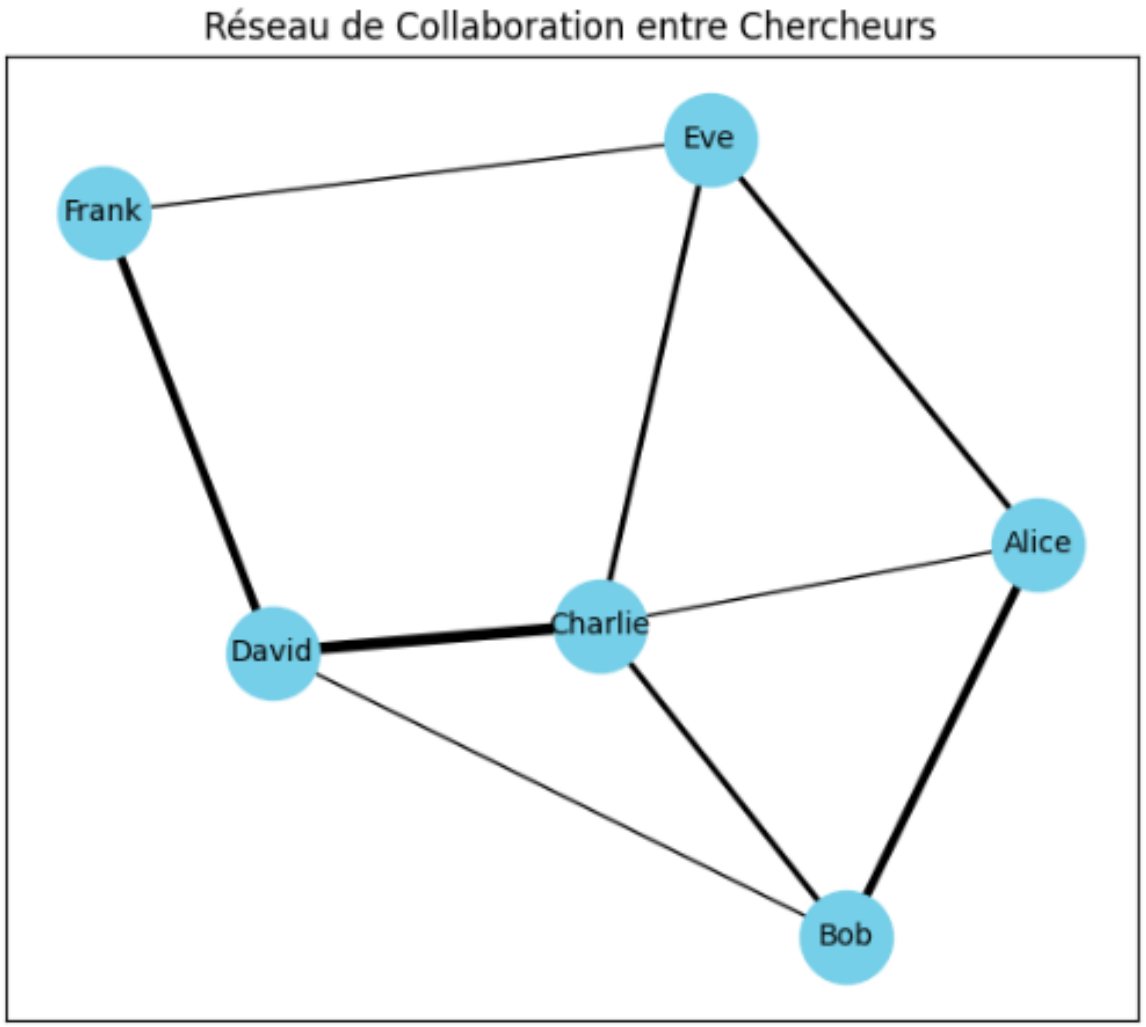
- Exemple : Réseau de Collaboration entre Chercheurs.

	Chercheur 1	Chercheur 2	Nombre de publications ensemble
0	Alice	Bob	3
1	Alice	Charlie	1
2	Bob	Charlie	2
3	Bob	David	1
4	Charlie	David	4
5	Alice	Eve	2
6	Eve	Frank	1
7	David	Frank	3
8	Charlie	Eve	2

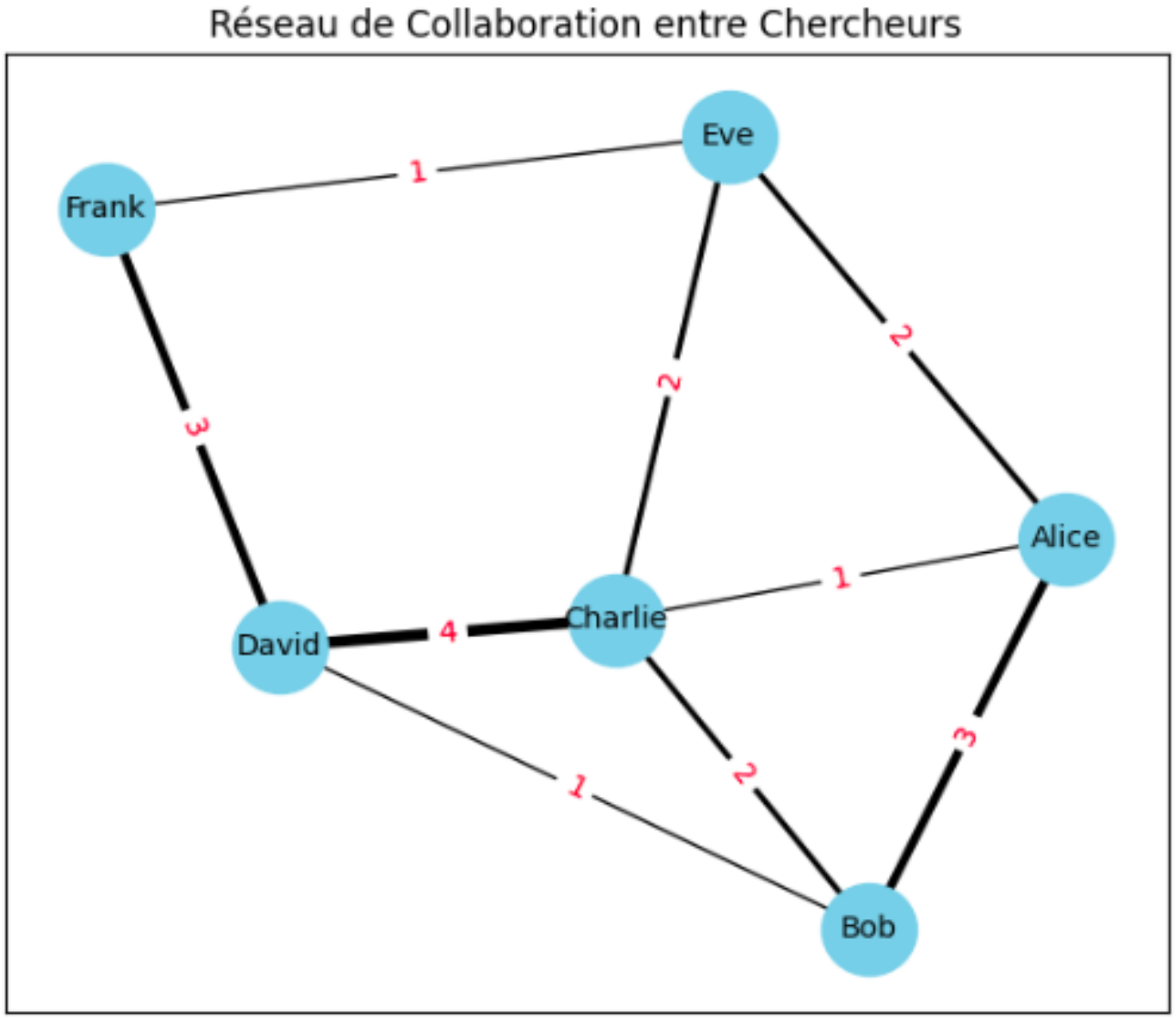
```
import networkx as nx
import matplotlib.pyplot as plt
# Jeu de données des collaborations
data = [ ('Alice', 'Bob', 3), ('Alice', 'Charlie', 1), ('Bob', 'Charlie', 2), ('Bob', 'David', 1), ('Charlie', 'David', 4), ('Alice', 'Eve', 2), ('Eve', 'Frank', 1), ('David', 'Frank', 3), ('Charlie', 'Eve', 2) ]
# Création du graphe
G = nx.Graph()
# Ajout des arêtes et des poids (nombre de publications)
for chercheur1, chercheur2, publications in data:
    G.add_edge(chercheur1, chercheur2, weight=publications)
# Définir la position des nœuds pour une disposition claire
pos = nx.spring_layout(G)
# Dessiner le graphe
plt.figure(figsize=(10, 8))
nx.draw_networkx_nodes(G, pos, node_size=1000, node_color='skyblue')
nx.draw_networkx_labels(G, pos, font_size=10, font_family="sans-serif")
plt.title("Réseau de Collaboration entre Chercheurs")
plt.show()
```

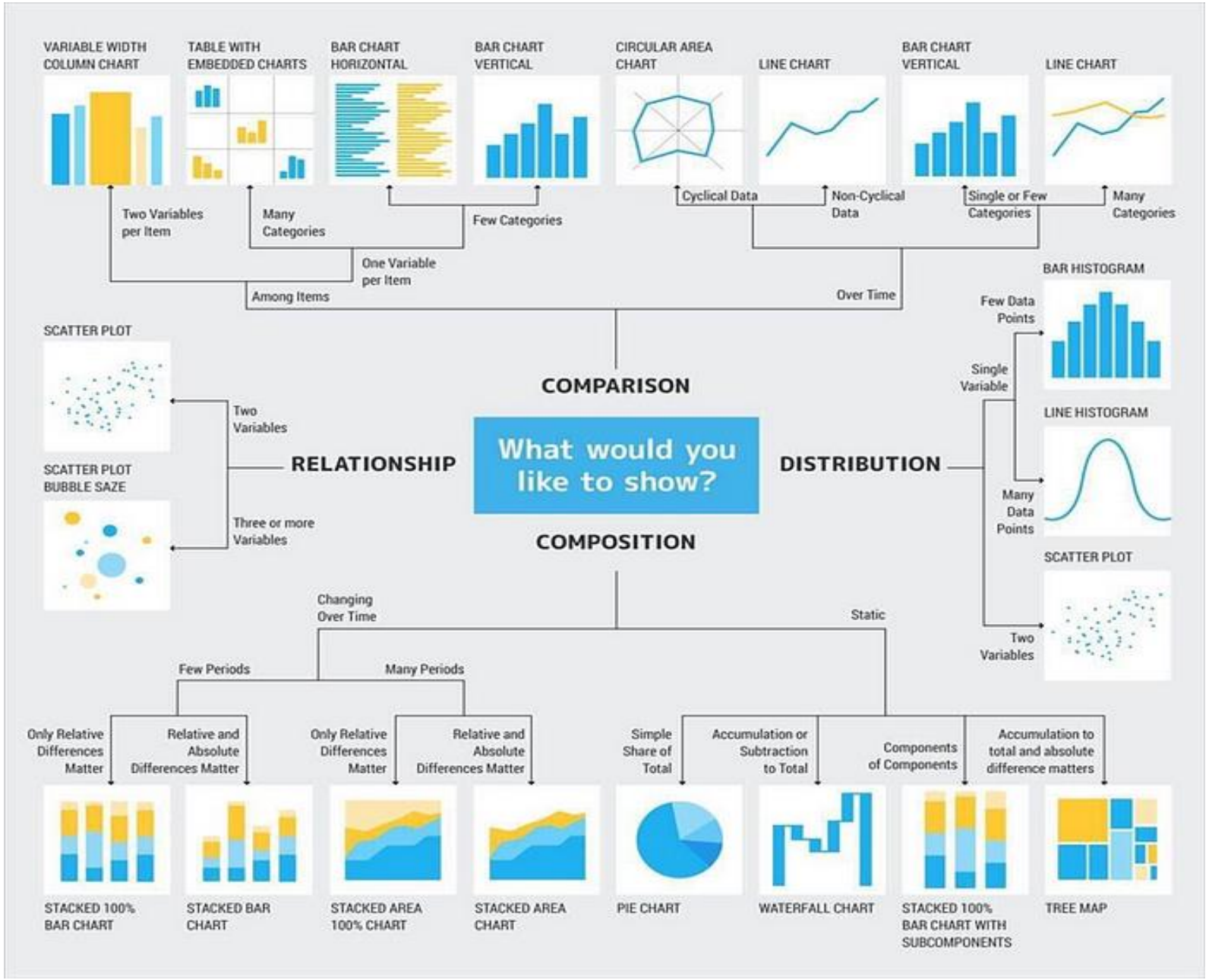

Réseau de Collaboration entre Chercheurs
Eve
Frank
Alice
Charlie
David
Bob

```
# Dessiner les arêtes avec des largeurs en fonction du nombre de publications
edges = nx.draw_networkx_edges(G, pos, width=[G[u][v]['weight'] for u, v in G.edges()])
# Ajouter une étiquette pour le poids de chaque arête (optionnel)
edge_labels = {(u, v): f"{G[u][v]['weight']}" for u, v in G.edges()}
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_color='red')

plt.title("Réseau de Collaboration entre Chercheurs")
plt.show()
```


Réseau de Collaboration entre Chercheurs
Eve
Frank
Alice
Charlie
David
Bob

```
# Ajouter une étiquette pour le poids de chaque arête (optionnel)
edge_labels = {(u, v): f"{G[u][v]['weight']}" for u, v in G.edges()}
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_color='red')

plt.title("Réseau de Collaboration entre Chercheurs")
plt.show()
```


Réseau de Collaboration entre Chercheurs
Eve
Frank
Alice
Charlie
David
Bob
1
2
2
3
4
1
1
2
3


What would you like to show?
COMPARISON
RELATIONSHIP
DISTRIBUTION
COMPOSITION
Two Variables per Item
Many Categories
Few Categories
One Variable per Item
Among Items
Cyclical Data
Non-Cyclical Data
Single or Few Categories
Many Categories
Over Time
VARIABLE WIDTH COLUMN CHART
TABLE WITH EMBEDDED CHARTS
BAR CHART HORIZONTAL
BAR CHART VERTICAL
CIRCULAR AREA CHART
LINE CHART
BAR CHART VERTICAL
LINE CHART
SCATTER PLOT
Two Variables
SCATTER PLOT BUBBLE SAZE
Three or more Variables
BAR HISTOGRAM
Few Data Points
Single Variable
LINE HISTOGRAM
Many Data Points
SCATTER PLOT
Two Variables
Changing Over Time
Static
Few Periods
Many Periods
Only Relative Differences Matter
Relative and Absolute Differences Matter
Only Relative Differences Matter
Relative and Absolute Differences Matter
Simple Share of Total
Accumulation or Subtraction to Total
Components of Components
Accumulation to total and absolute difference matters
STACKED 100% BAR CHART
STACKED BAR CHART
STACKED AREA 100% CHART
STACKED AREA CHART
PIE CHART
WATERFALL CHART
STACKED 100% BAR CHART WITH SUBCOMPONENTS
TREE MAP

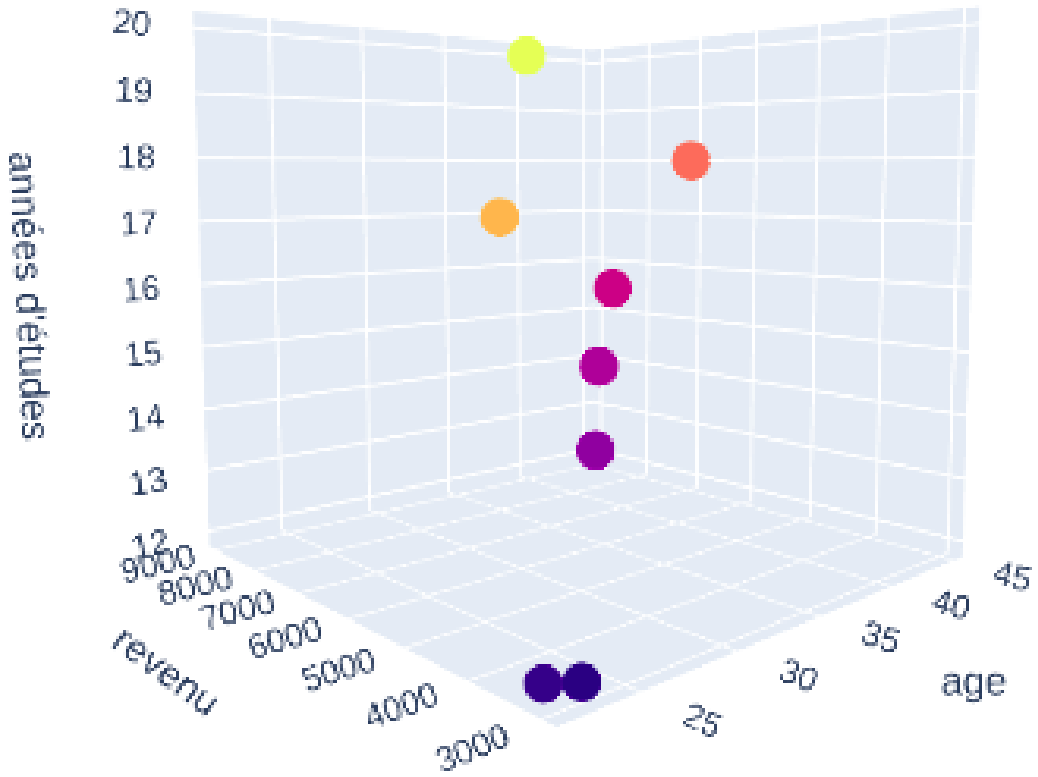
1. PRÉREQUIS de la compétence
2. Méthodes de nettoyage des données
    - Traitement des valeurs manquantes
    - Normalisation et standardisation
    - Codage des variables catégorielles
3. Analyse exploratoire des données
    - Analyse univariée
    - Analyse bivariée et corrélation
4. Techniques de visualisation de données
    - Maîtriser la visualisation avancée des données
    - Graphiques 3D et interactifs
    - Visualisations complexes

Maîtriser la visualisation avancée des données:

1. **Graphiques 3D et interactifs** : Utile pour explorer des relations entre trois variables.

- Exemple : les revenus en fonction de l'age et les années d'études.

age	revenu	années d'études
23	3000	12
45	7000	18
31	5000	15
35	8000	17
40	9000	20
29	4500	14
22	3200	12
34	5600	16




```
import plotly.express as px
import pandas as pd

# Exemple de données
data = pd.DataFrame({
'age': [23, 45, 31, 35, 40, 29, 22, 34],
'revenu': [3000, 7000, 5000, 8000, 9000, 4500, 3200, 5600],
'années d\'études': [12, 18, 15, 17, 20, 14, 12, 16] })

# Création d'un nuage de points 3D interactif
fig = px.scatter_3d(data, x='age', y='revenu', z='années d\'études', color='revenu', title='Relation entre Âge, Revenu et Années d\'études')

fig.show()
```

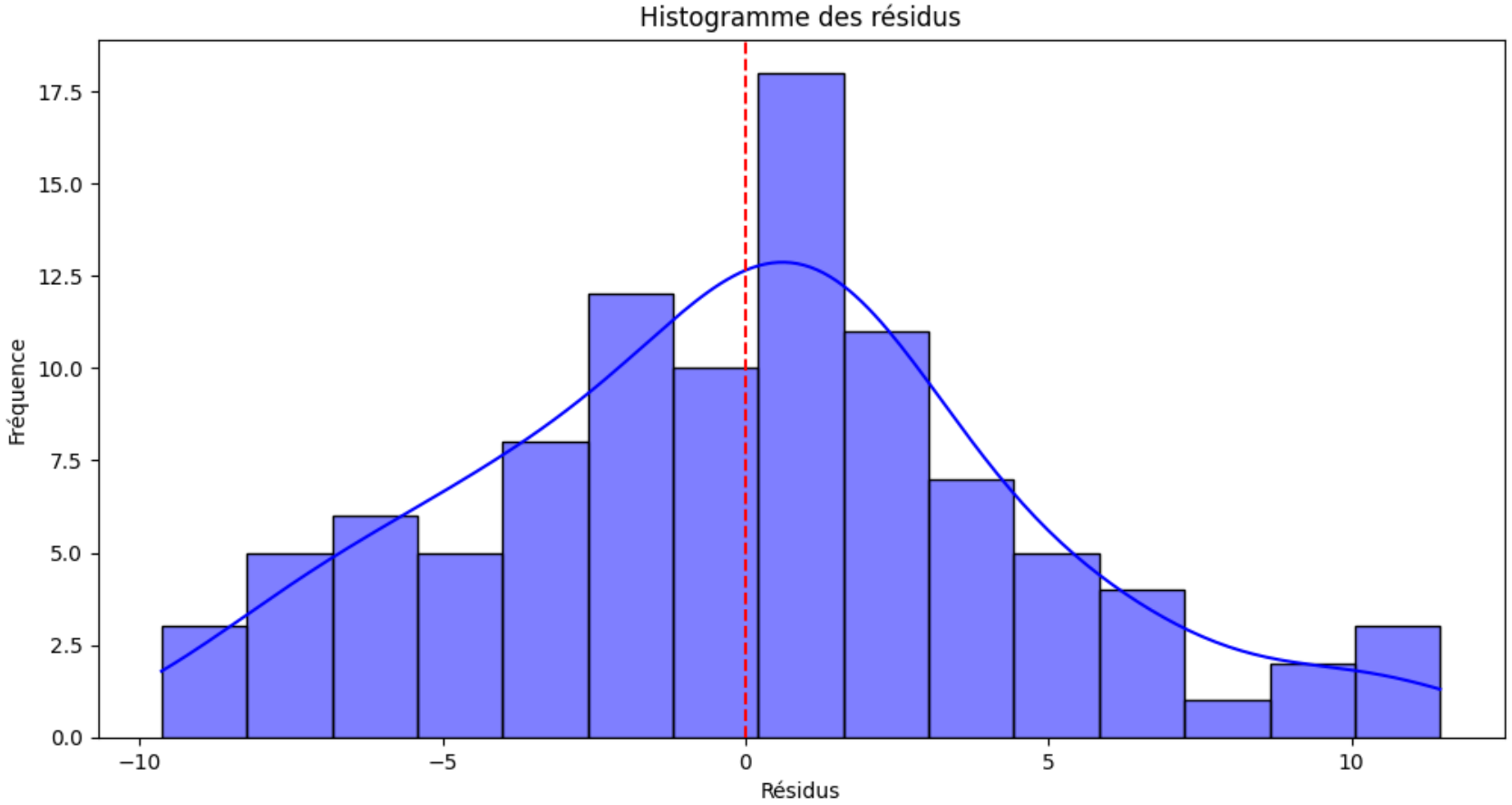
1. PRÉREQUIS de la compétence
2. Méthodes de nettoyage des données
    - Traitement des valeurs manquantes
    - Normalisation et standardisation
    - Codage des variables catégorielles
3. Analyse exploratoire des données
    - Analyse univariée
    - Analyse bivariée et corrélation
4. Techniques de visualisation de données
    - Maîtriser la visualisation avancée des données
    - Graphiques 3D et interactifs
    - Visualisations complexes

Les hypothèses des modèles linéaires

Un modèle linéaire classique repose sur les hypothèses suivantes :

- **Normalité des résidus** : Les erreurs doivent suivre une distribution normale.
- **Homoscédasticité** : La variance des résidus doit être constante.

Normalité des résidus

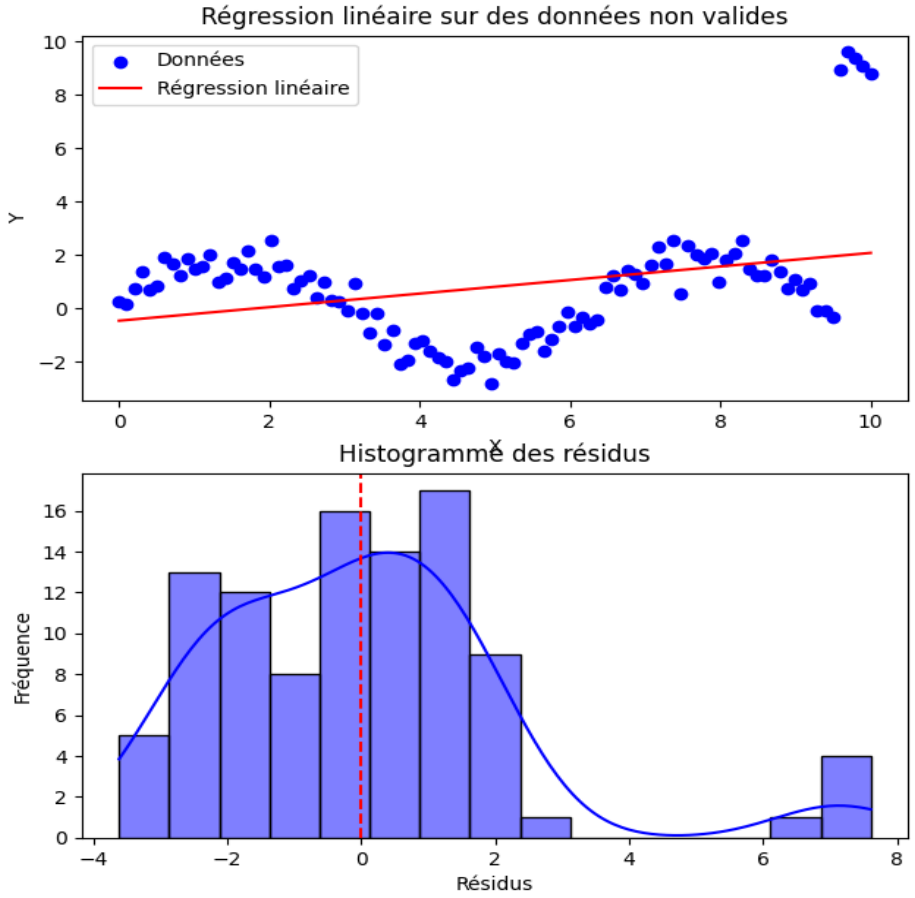


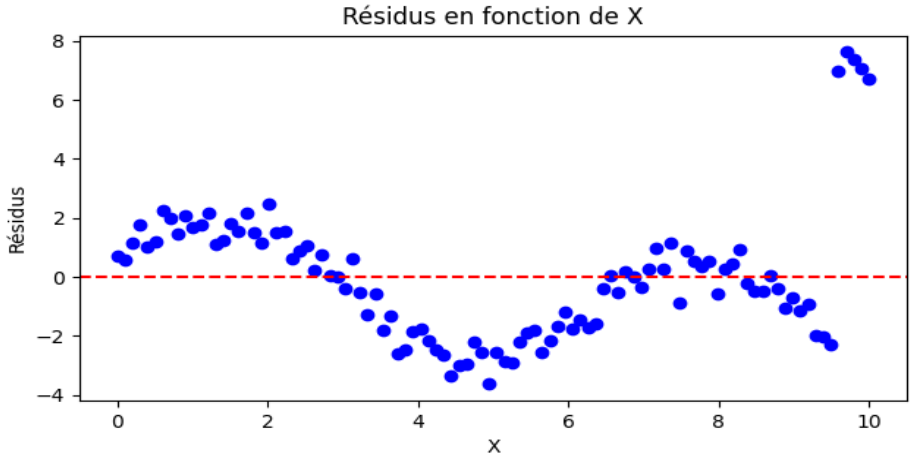
Une courbe en cloche centrée sur zéro est un bon indicateur de normalité.

Normalité des résidus

- Exemple jeu de données: cas de régression de la fonction 2*sin(x)

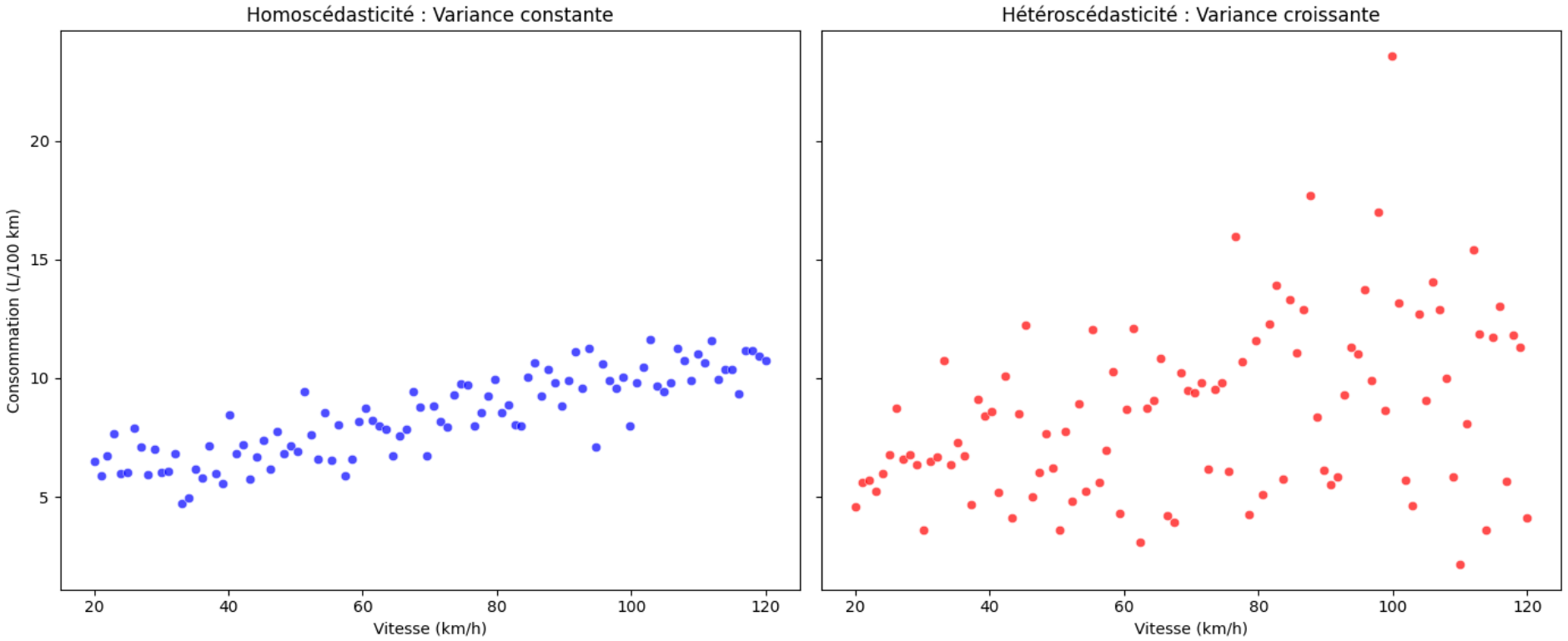
	X	Y
0	0.000000	0.248357
1	0.101010	0.132545
2	0.202020	0.725142
3	0.303030	1.358343
4	0.404040	0.669197
5	0.505051	0.850635
6	0.606061	1.928875
7	0.707071	1.682936
8	0.808081	1.211188
9	0.909091	1.849171





Les résidus ne suivent pas une distribution normale. ==> régression linéaire n'est pas valide

Homoscédasticité



Homoscédasticité : Les erreurs ont une variance constante indépendamment de la vitesse.
Hétéroscédasticité : La variance des erreurs augmente avec la vitesse.==> **régression linéaire n'est pas valide**

5. Analyser les données Time Series (Forcasting)

- Introduction
- Que peut-on prévoir ?
- Technique d'analyse Time Series
- Méthode de prédiction (forcasting) ARIMA
- Méthode de prédiction (forcasting) SARIMA
- Méthode de prédiction (forcasting) LSTM

5. Analyser les données Time Series (Forcasting)

- Introduction
- Que peut-on prévoir ?
- Technique d'analyse Time Series
- Méthode de prédiction (forcasting) ARIMA
- Méthode de prédiction (forcasting) SARIMA
- Méthode de prédiction (forcasting) LSTM

Introduction

La prévision est nécessaire dans de nombreuses situations :

- planifier le personnel dans un centre d'appels pour la semaine prochaine exige des prévisions du volume d'appels
- gérer un stock nécessite des prévisions des besoins en inventaire.
- Prévoir les conditions météorologiques pour optimiser les récoltes ou prévenir les catastrophes naturelles.
- Les prévisions peuvent être nécessaires:
  - plusieurs années à l'avance (pour les investissements en capital)
  - seulement quelques minutes à l'avance (pour le routage des télécommunications).

==> Quelles que soient les circonstances ou les horizons temporels concernés, la prévision est un outil essentiel pour une planification efficace et efficiente.

Qu'Est-ce qu'on peut prédire au futur?

Certaines choses sont plus faciles à prévoir que d'autres. :

- L'heure du lever du soleil demain matin (très facile)
- Ventes mensuelles d'un produit de grande consommation (modérée)
- les numéros du loto de demain (très difficile)

==> Quelles sont les facteurs qui désignent qu'une prédiction soit facile ou difficile ?

Les facteurs de prédire de futur (forcasting)

La prédiction d'un événement ou d'une quantité dépend de quatre facteurs:

- La compréhension des facteurs qui y contribuent : Plus nous comprenons les facteurs qui influencent un phénomène, plus il est facile de le prévoir.
- La quantité de données disponibles : Plus il y a de données historiques, plus il est possible de construire des modèles de prévision précis.
- La similarité entre le futur et le passé : Si le futur est susceptible de ressembler au passé, les prévisions seront plus fiables.
- L'impact des prévisions sur ce que l'on cherche à prévoir : Parfois, les prévisions elles-mêmes peuvent influencer le résultat, ce qui rend la prévision plus complexe.

Les facteurs de prédire de futur (forcasting)

La prédiction d'un événement ou d'une quantité dépend de quatre facteurs:

- La compréhension des facteurs qui y contribuent : Plus nous comprenons les facteurs qui influencent un phénomène, plus il est facile de le prévoir.
- La quantité de données disponibles : Plus il y a de données historiques, plus il est possible de construire des modèles de prévision précis.
- La similarité entre le futur et le passé : Si le futur est susceptible de ressembler au passé, les prévisions seront plus fiables.
- L'impact des prévisions sur ce que l'on cherche à prévoir : Parfois, les prévisions elles-mêmes peuvent influencer le résultat, ce qui rend la prévision plus complexe.

Exemple:

les prévisions à court terme de la demande en électricité résidentielle peuvent être très précises, car ces quatre conditions sont généralement remplies :

- Compréhension des facteurs : La demande en électricité est largement influencée par les températures, avec des effets secondaires liés aux variations calendaires (comme les vacances) et aux conditions économiques.
- Disponibilité des données : Il existe généralement plusieurs années de données sur la demande en électricité et des décennies de données sur les conditions météorologiques.
- Similarité entre le futur et le passé : Pour des prévisions à court terme (jusqu'à quelques semaines), on peut supposer que le comportement de la demande sera similaire à ce qui a été observé par le passé.
- Impact des prévisions : Pour la plupart des utilisateurs résidentiels, le prix de l'électricité ne dépend pas de la demande, donc les prévisions de demande n'ont que peu ou pas d'effet sur le comportement des consommateurs.

==>si ces facteurs sont validés on peut dire que la prédiction est facile?

Exercice1 :

En tant que Data Analyst, une entreprise X vous a demandé d'analyser s'il est possible de concevoir un modèle de prédiction pour les taux de change des devises.

1. Quel est le type de ce problème?
2. Quels sont les facteurs valides pour la prédiction des taux de change des devises? Justifiez votre réponse.
3. qu'elle votre réponse à la demande de l'entreprise.

Exercice2:

Classez les prévisions futur (forcasting) ci-dessous, le plus facile au plus difficile. Justifiez votre réponse en vérifiant les quatre facteurs:

- Lever du soleil.
- Numéros gagnants du loto.
- Ventes mensuelles d'un produit de grande consommation.
- Prix des actions en bourse.
- Demande en électricité à court terme.

Time series forecasting

*" Tout ce qui est observé de manière **séquentielle au fil du temps** est une **série temporelle**."*

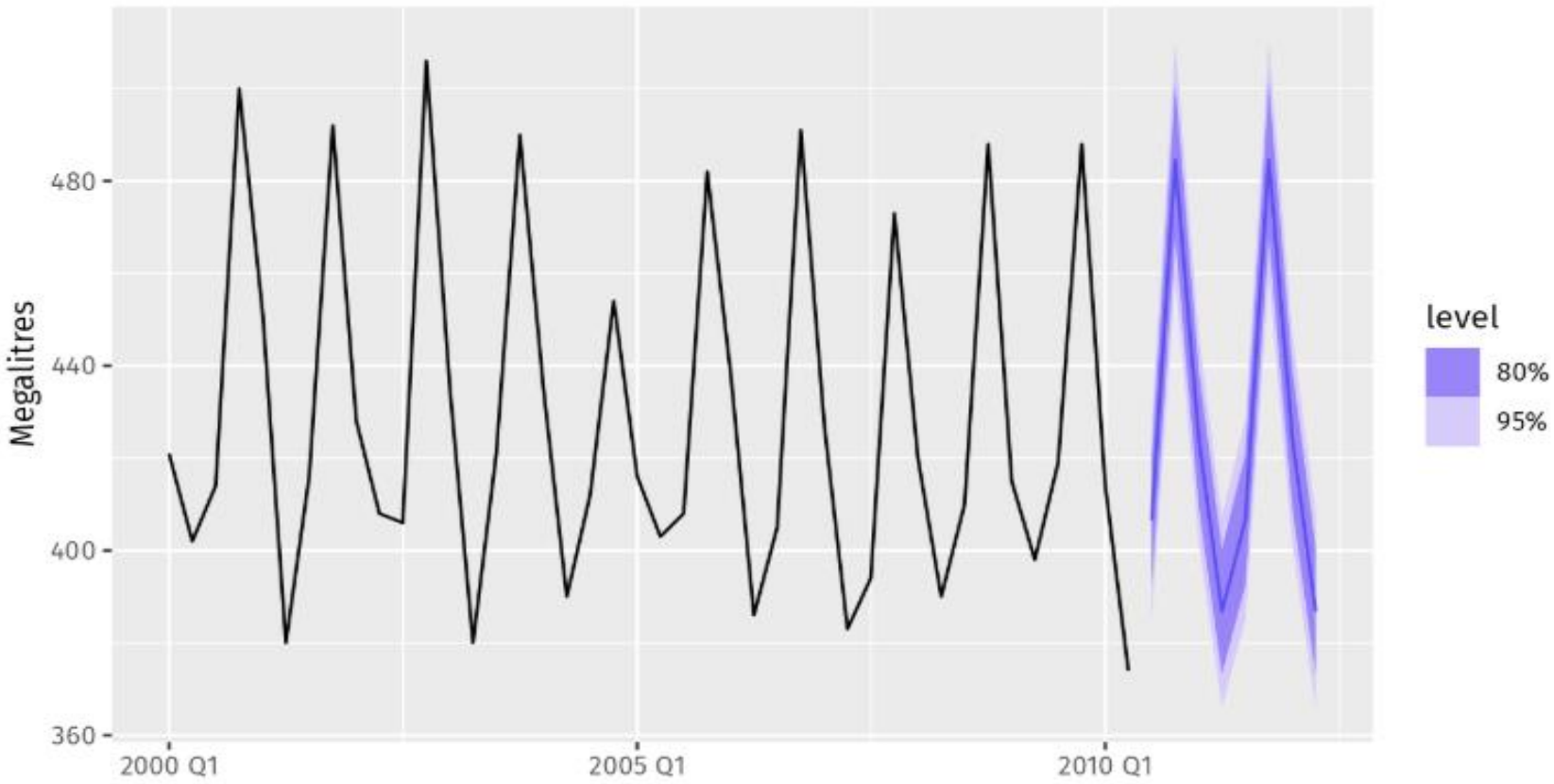
Des exemples de données de séries temporelles incluent :

- Les profits **annuels** de Google.
- Les résultats **trimestriels** des ventes d'Amazon.
- Les précipitations **mensuelles.**
- Les ventes au détail **hebdomadaires.**
- La demande en électricité **horaire.**

==> Dans ce cours, Nous ne considérerons que les séries temporelles observées à intervalles de temps réguliers (par exemple, toutes les heures, quotidiennement, hebdomadairement, mensuellement, trimestriellement, annuellement).

Time series forecasting

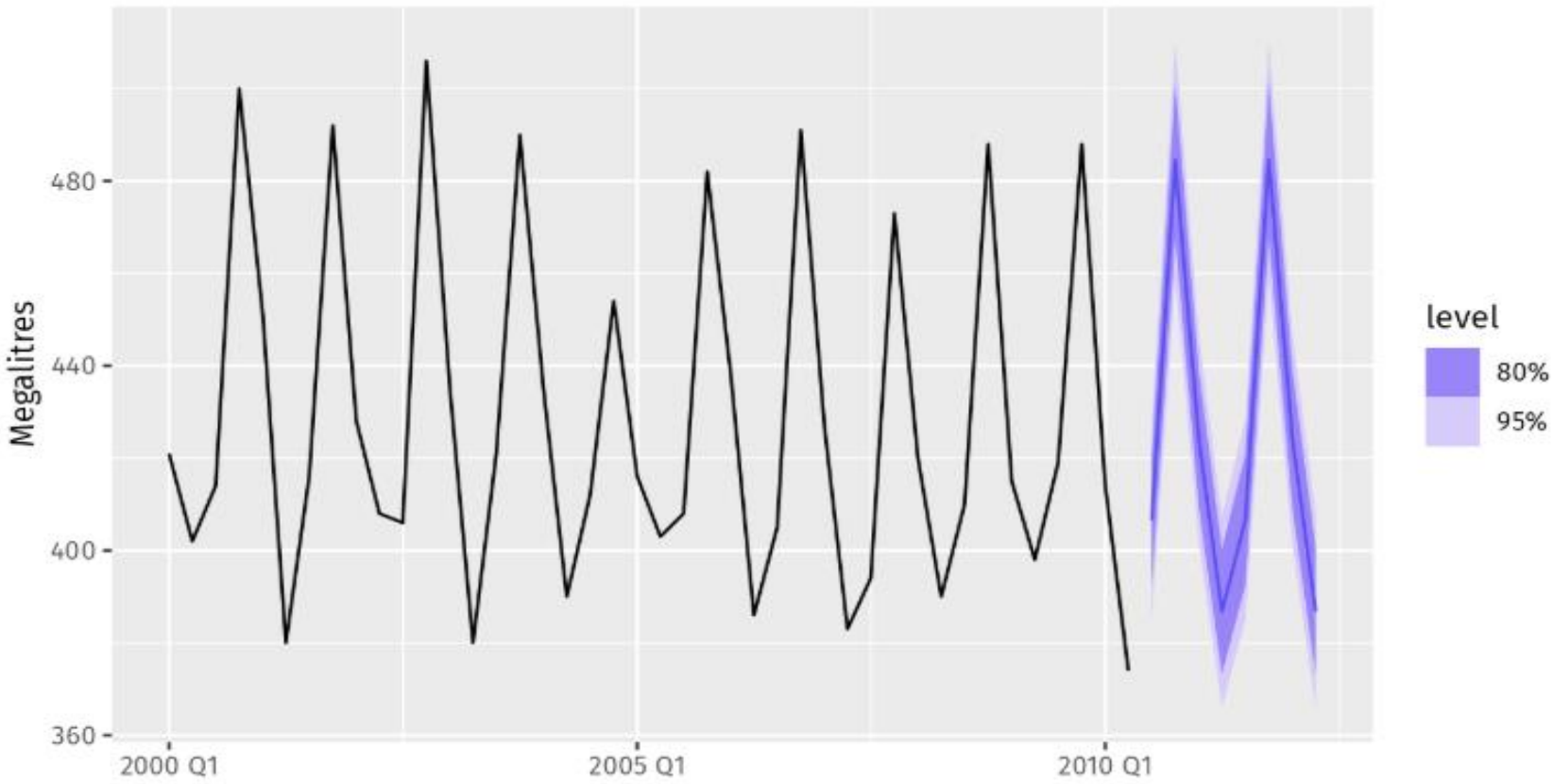
- Lors de la prévision des données de séries temporelles, l'objectif est **d'estimer comment la séquence d'observations évoluera dans le futur.**
- Quel est l'**intervalle de temps** régulier des valeurs observées au passé?
- Quel est l'**intervalle de temps** régulier des valeurs estimées au futur?



La figure montre la production de boissons. [2000, 2010+]

Time series forecasting

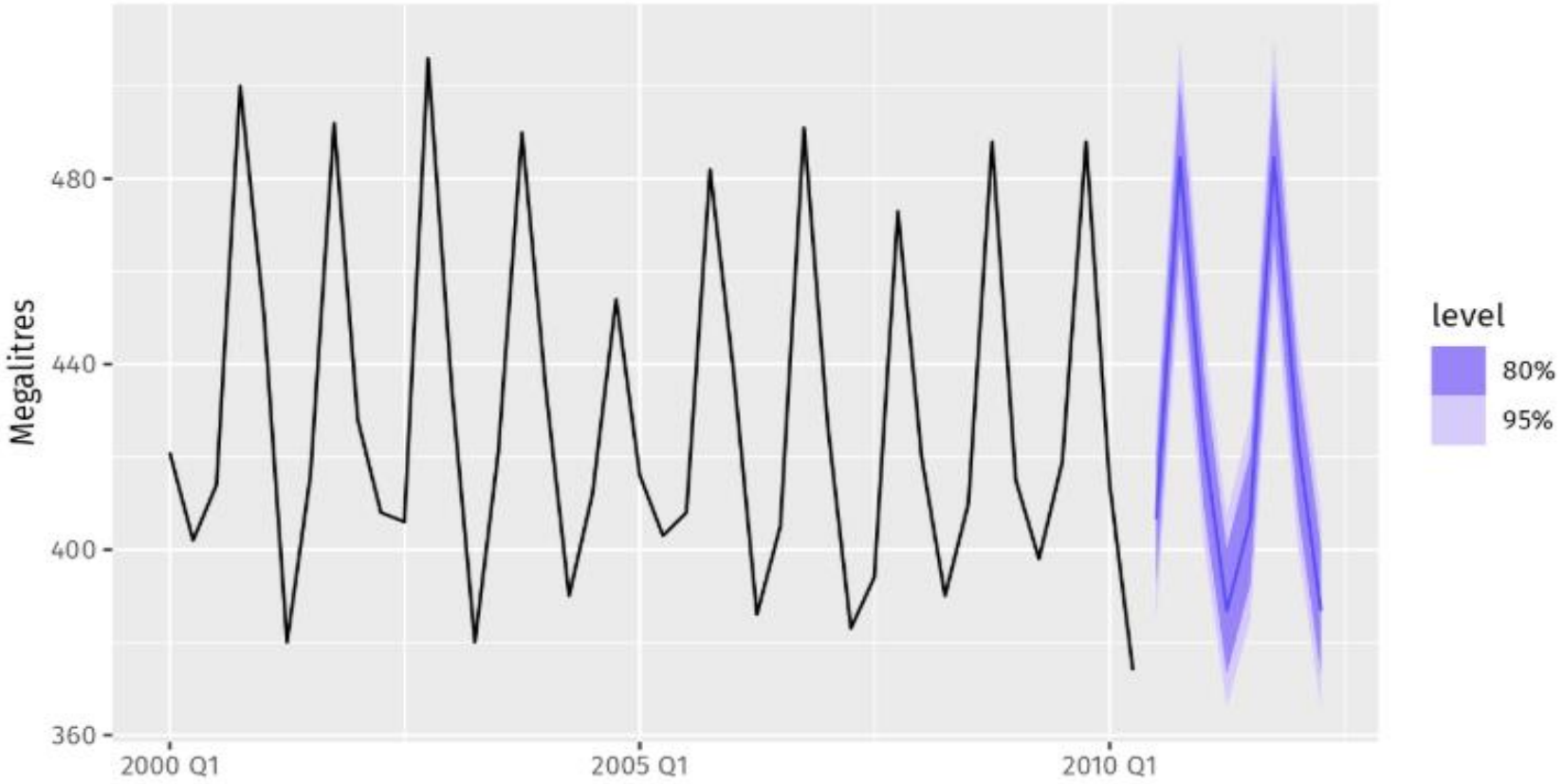
- Lors de la prévision des données de séries temporelles, l'objectif est **d'estimer comment la séquence d'observations évoluera dans le futur.**
- Quel est l'**intervalle de temps** régulier des valeurs observées au passé? trimestrielle
- Quel est l'**intervalle de temps** régulier des valeurs estimées au futur? 2 ans



La figure montre la production de boissons. [2000, 2010+]

Time series forecasting

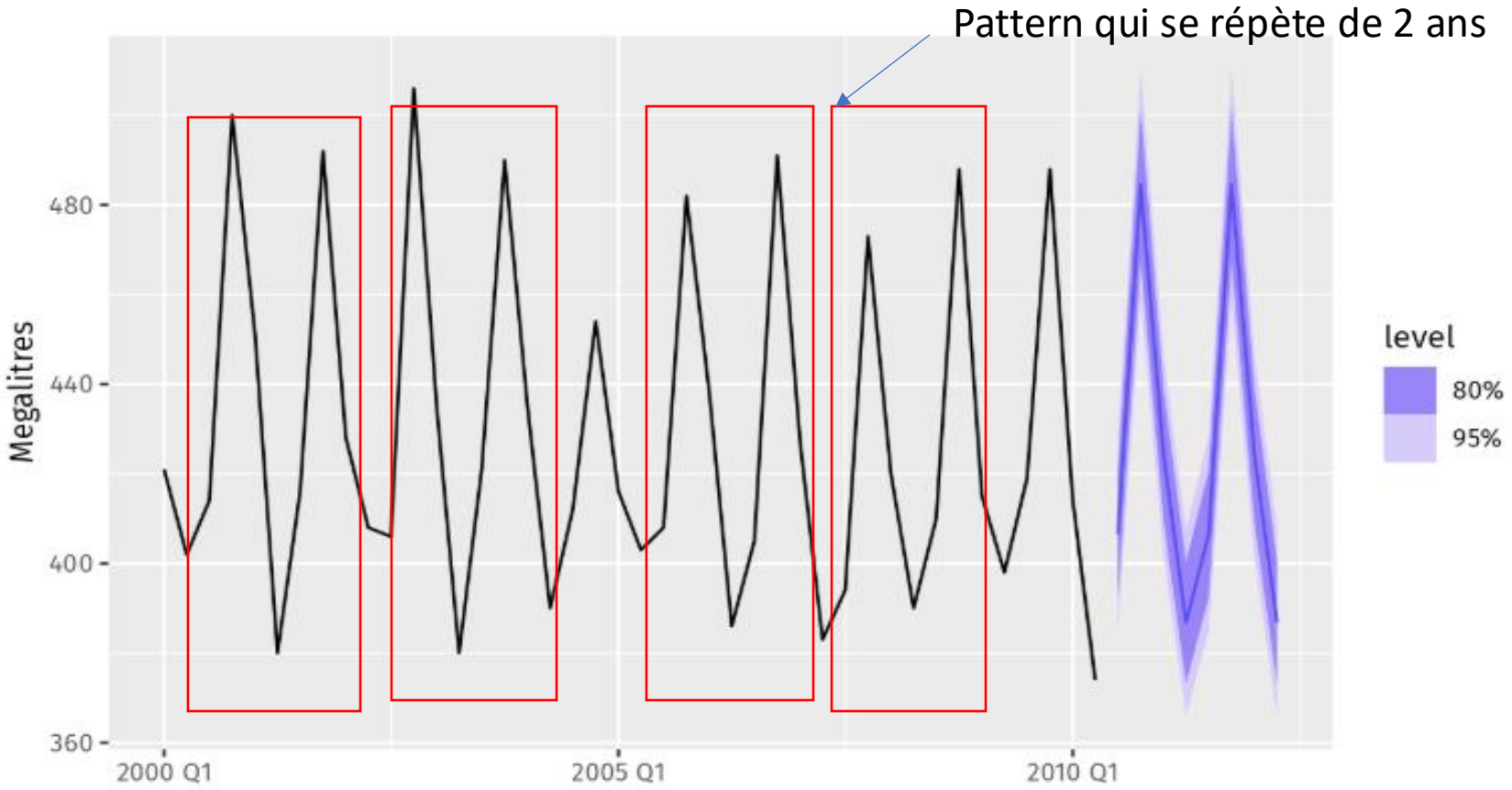
- Que peut-on conclure sur la prédiction (forcasting) selon la figure?



La figure montre la production de boissons. [2000, 2010+]

Time series forecasting

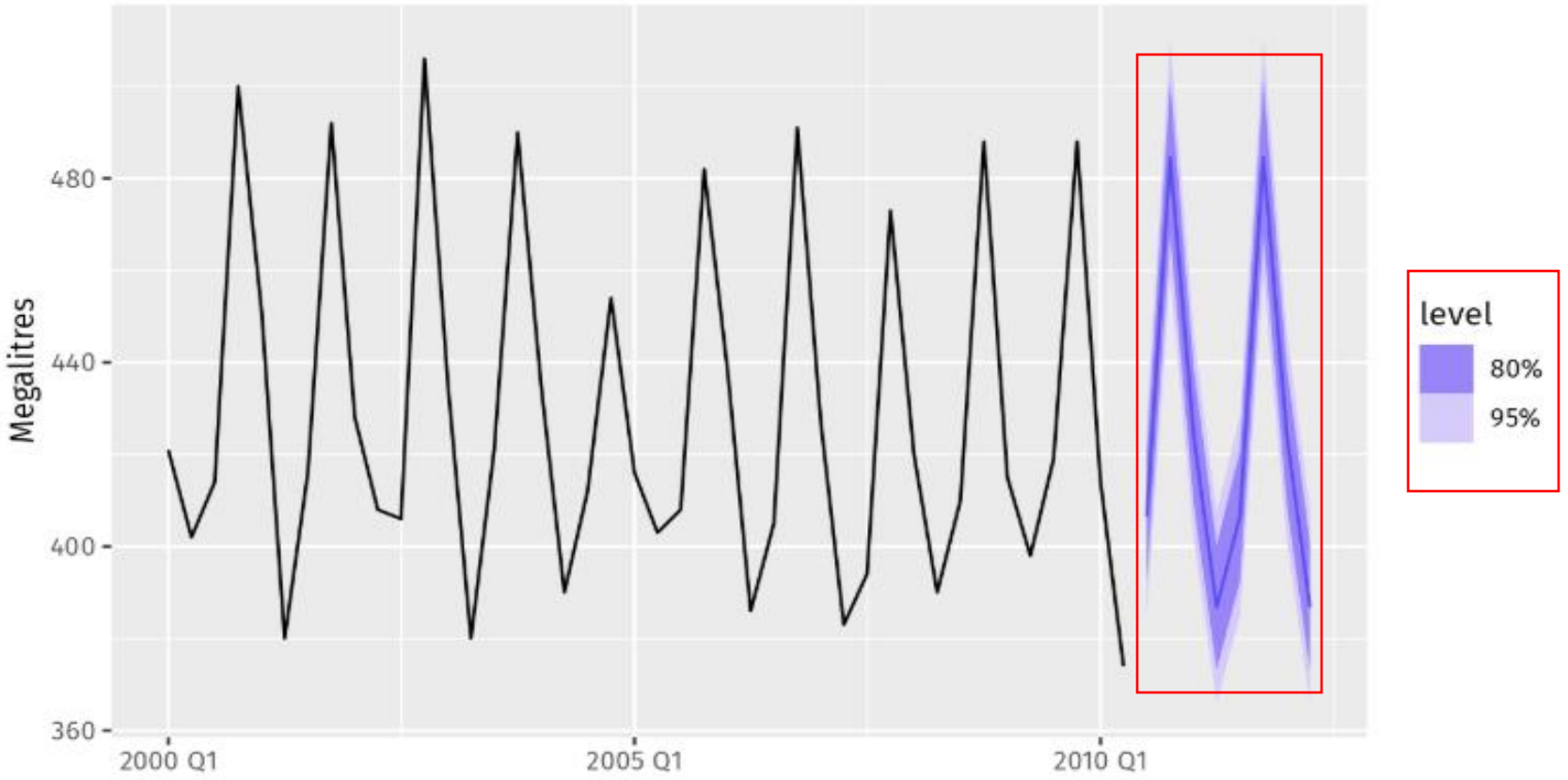
- Que peut-on conclure sur la prédiction (forcasting) selon la figure?
- Les prévisions ont capturé le modèle saisonnier (pattern) observé dans les données historiques et l'ont reproduit pour les deux prochaines années.



La figure montre la production de boissons. [2000, 2010+]

Time series forecasting

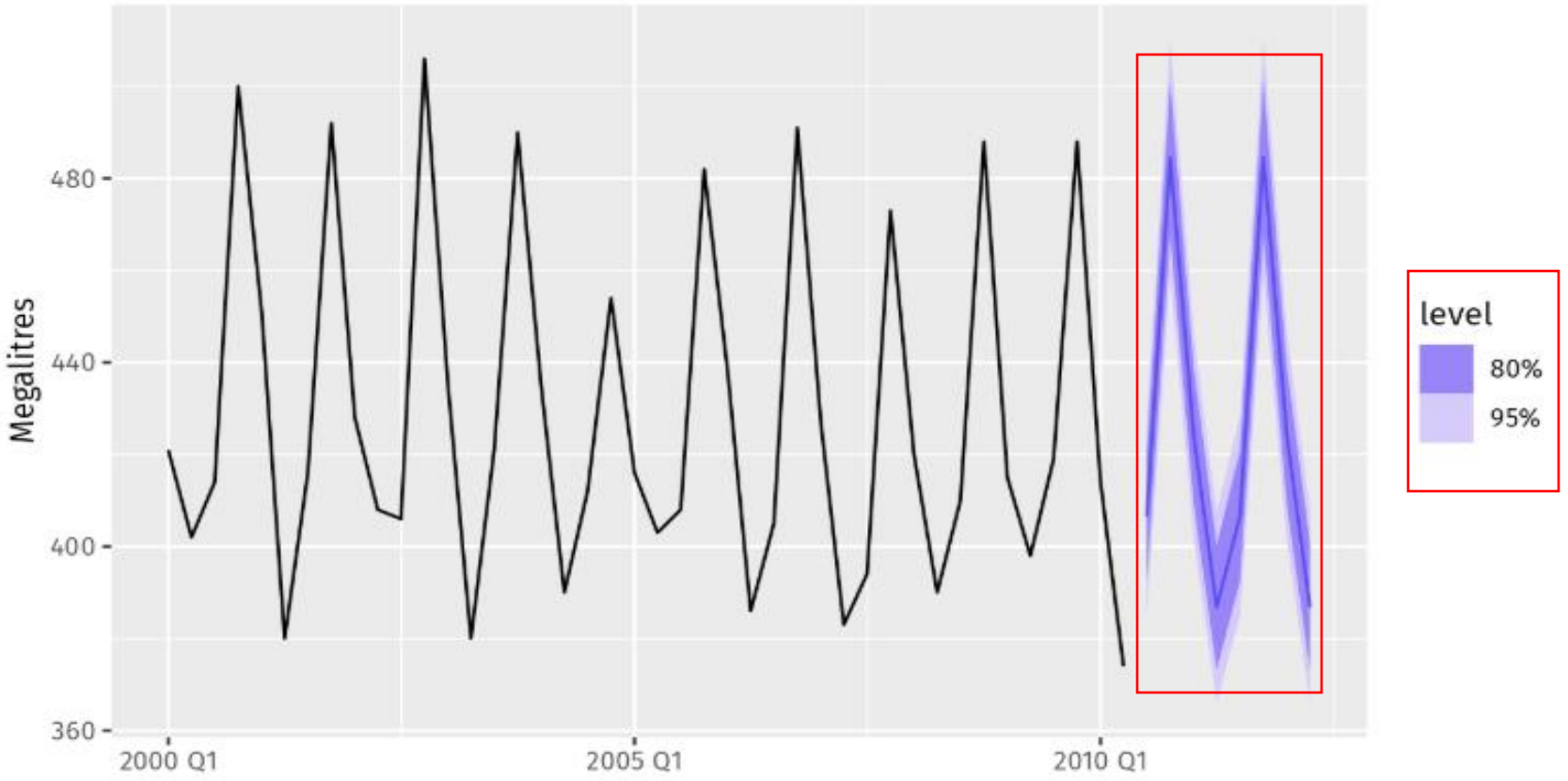
- Que représentent la **ligne blue**, la region **blue ombrée foncée** et la region **blue ombrée claire**?



La figure montre la production de boissons. [2000, 2010+]

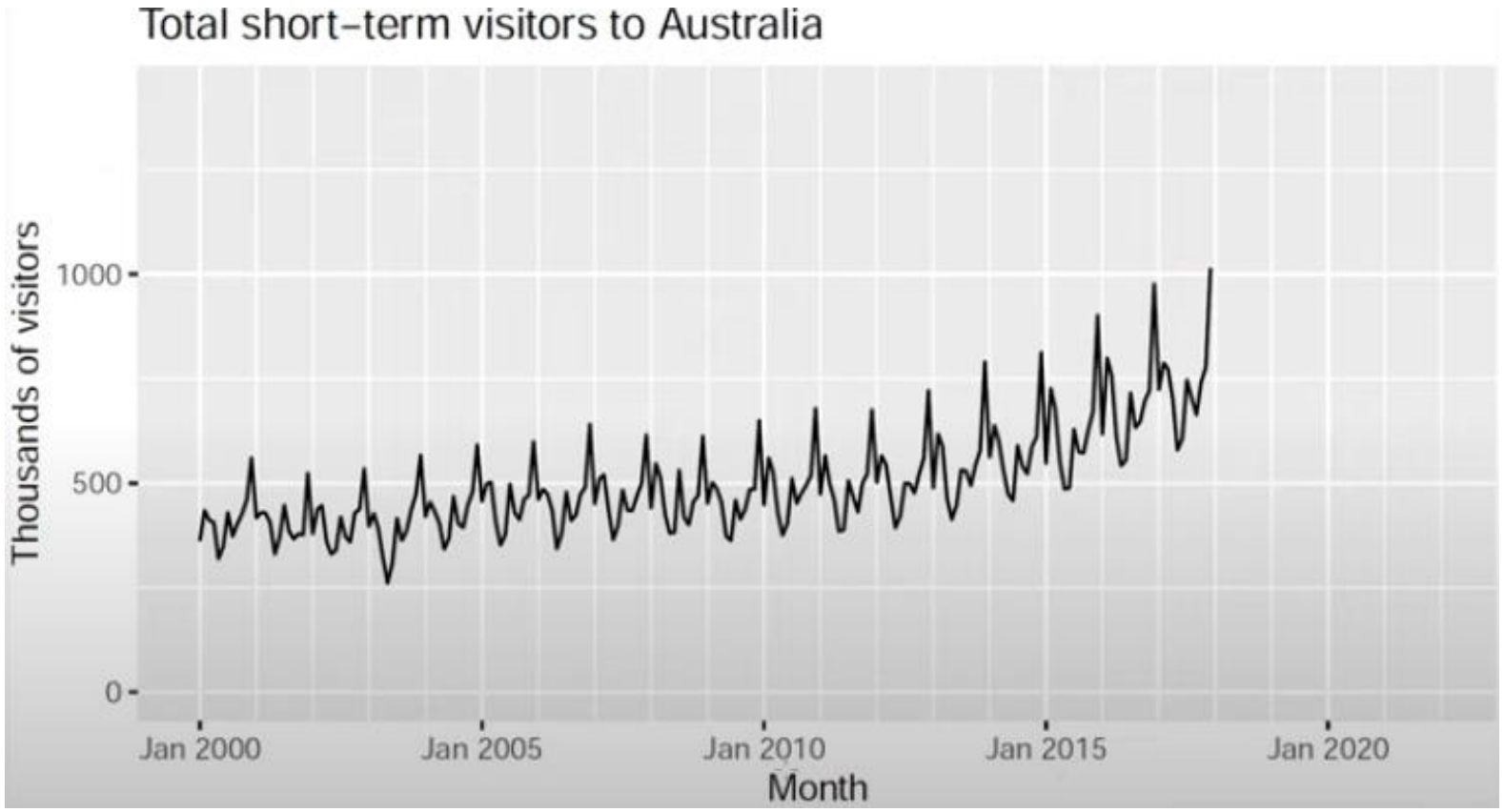
Time series forecasting

- Que représentent la **ligne blue**, la region **blue ombrée foncée** et la region **blue ombrée claire**?
- La région ombrée foncée représente les **intervalles de prédiction** à **80 %**.
- chaque valeur future est censée se situer dans cette région avec une probabilité de 80 %.
- La région ombrée claire représente les intervalles de prédiction à 95 %.
- Ces intervalles de prédiction sont un moyen utile de **visualiser l'incertitude** des prévisions.
- Dans ce cas, les prévisions sont supposées être **précises**, ce qui explique pourquoi les intervalles de prédiction sont relativement **étroits**.



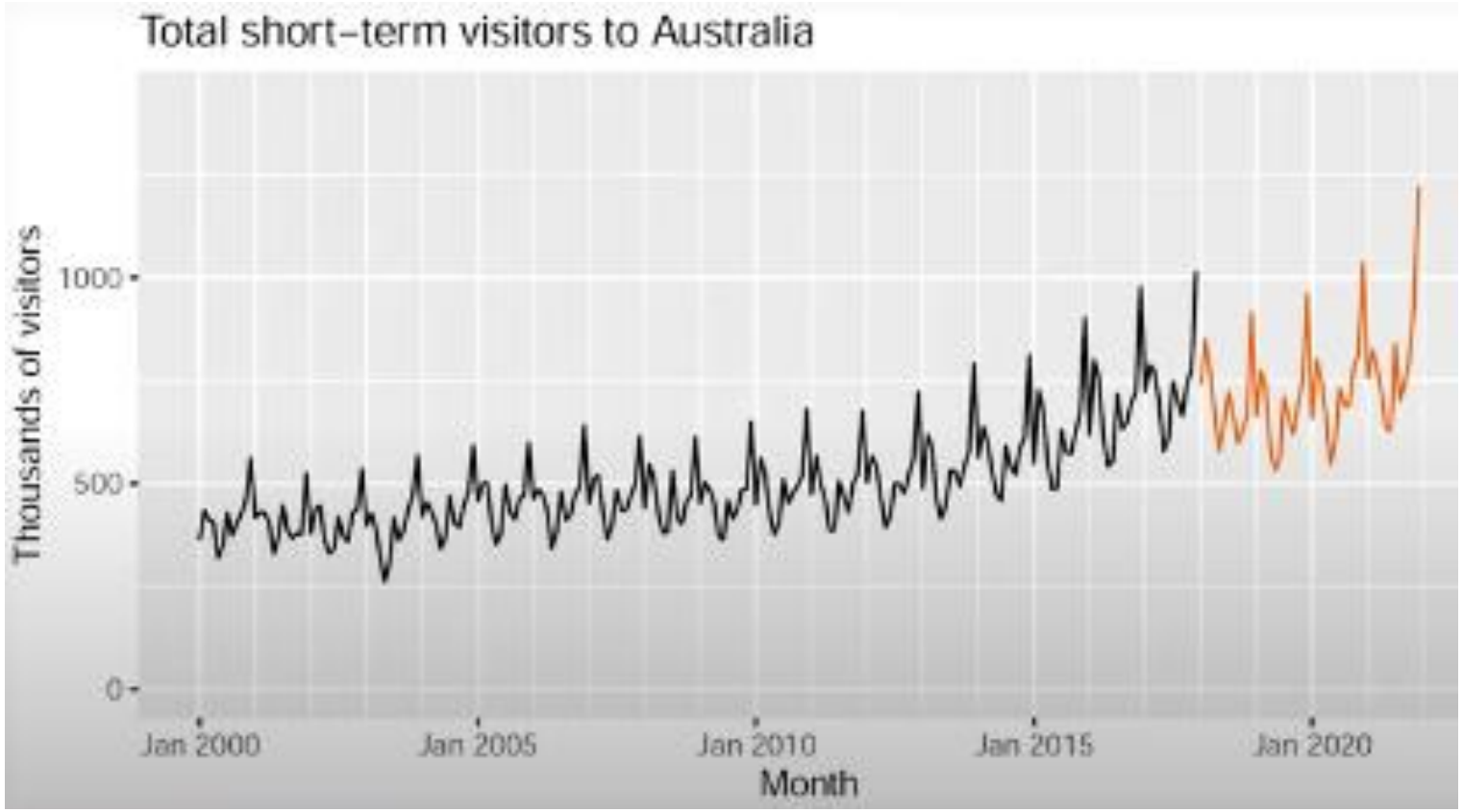
La figure montre la production de boissons. [2000, 2010+]

Time series forecasting



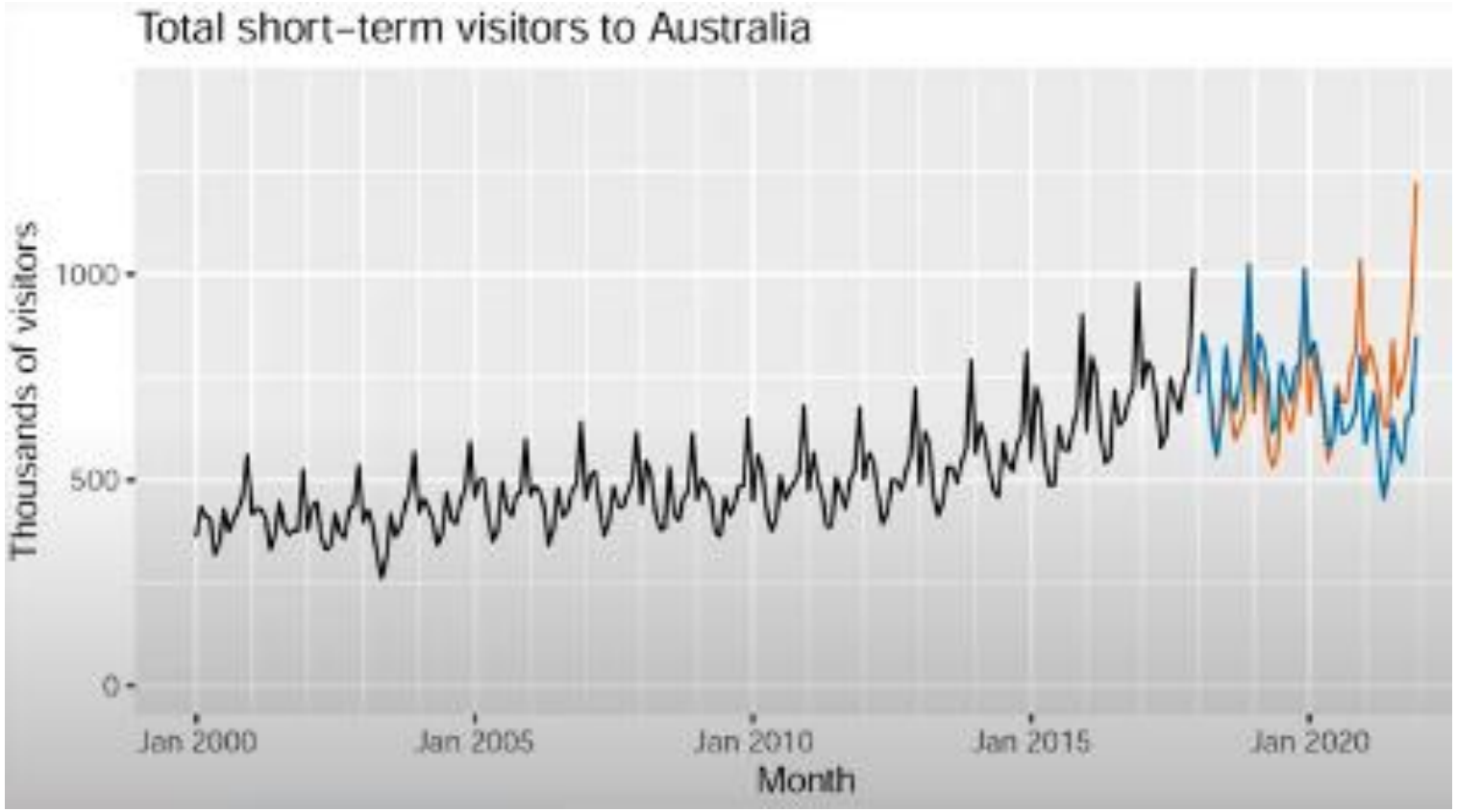
le nombre de visiteurs en australie. [2000, 2018]

Time series forecasting



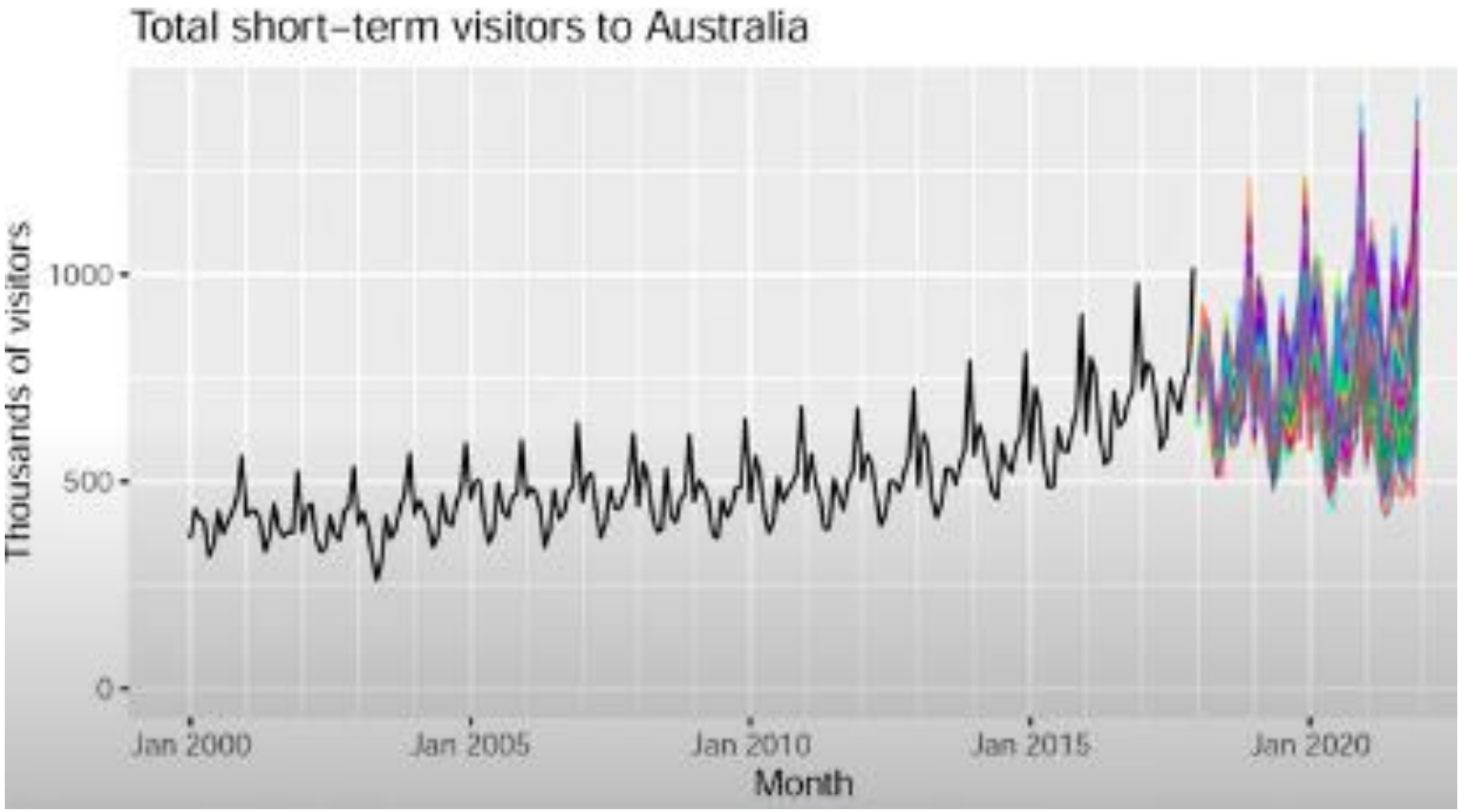
le nombre de visiteurs en australie. [2000, 2018]

Time series forecasting



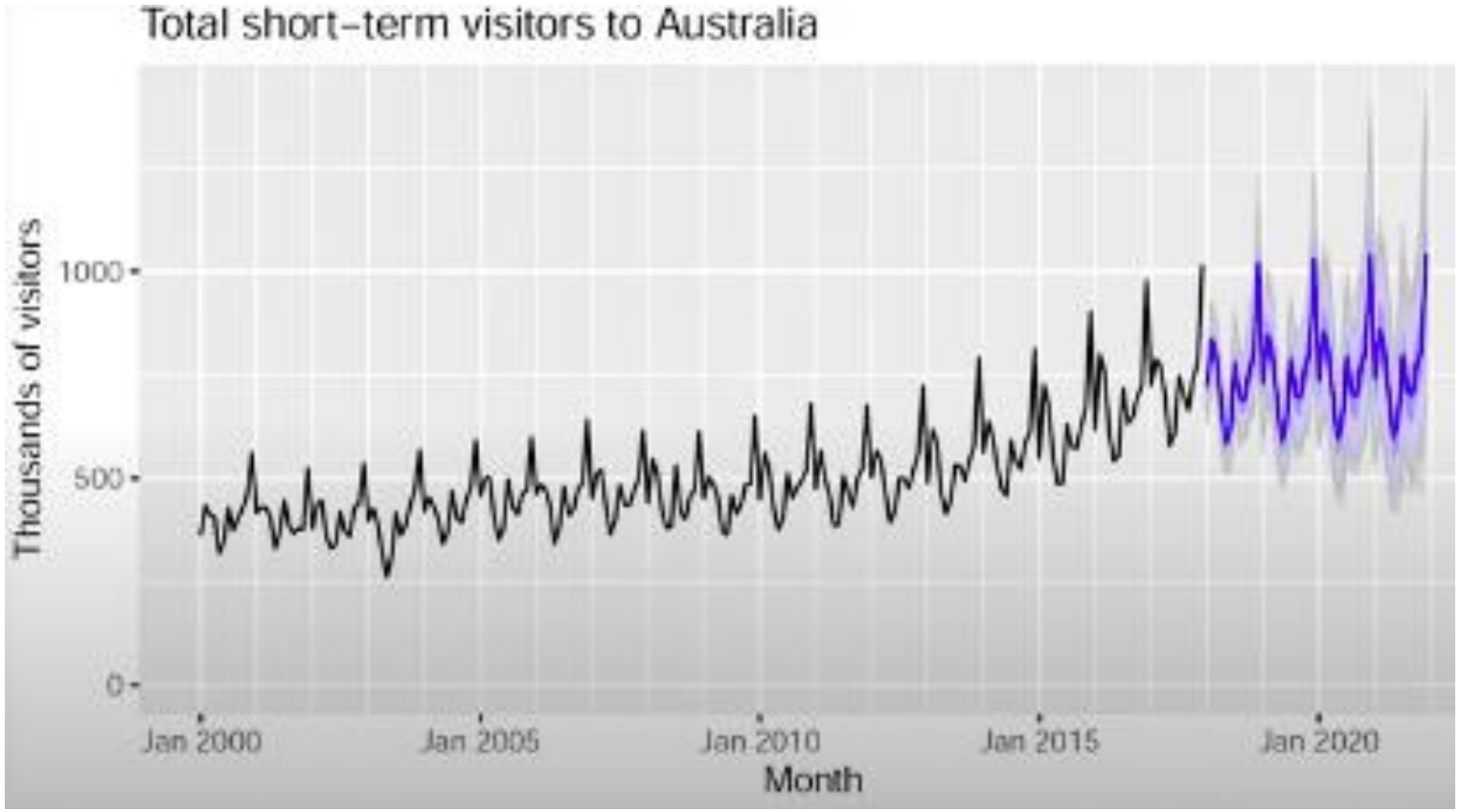
le nombre de visiteurs en australie. [2000, 2018]

Time series forecasting



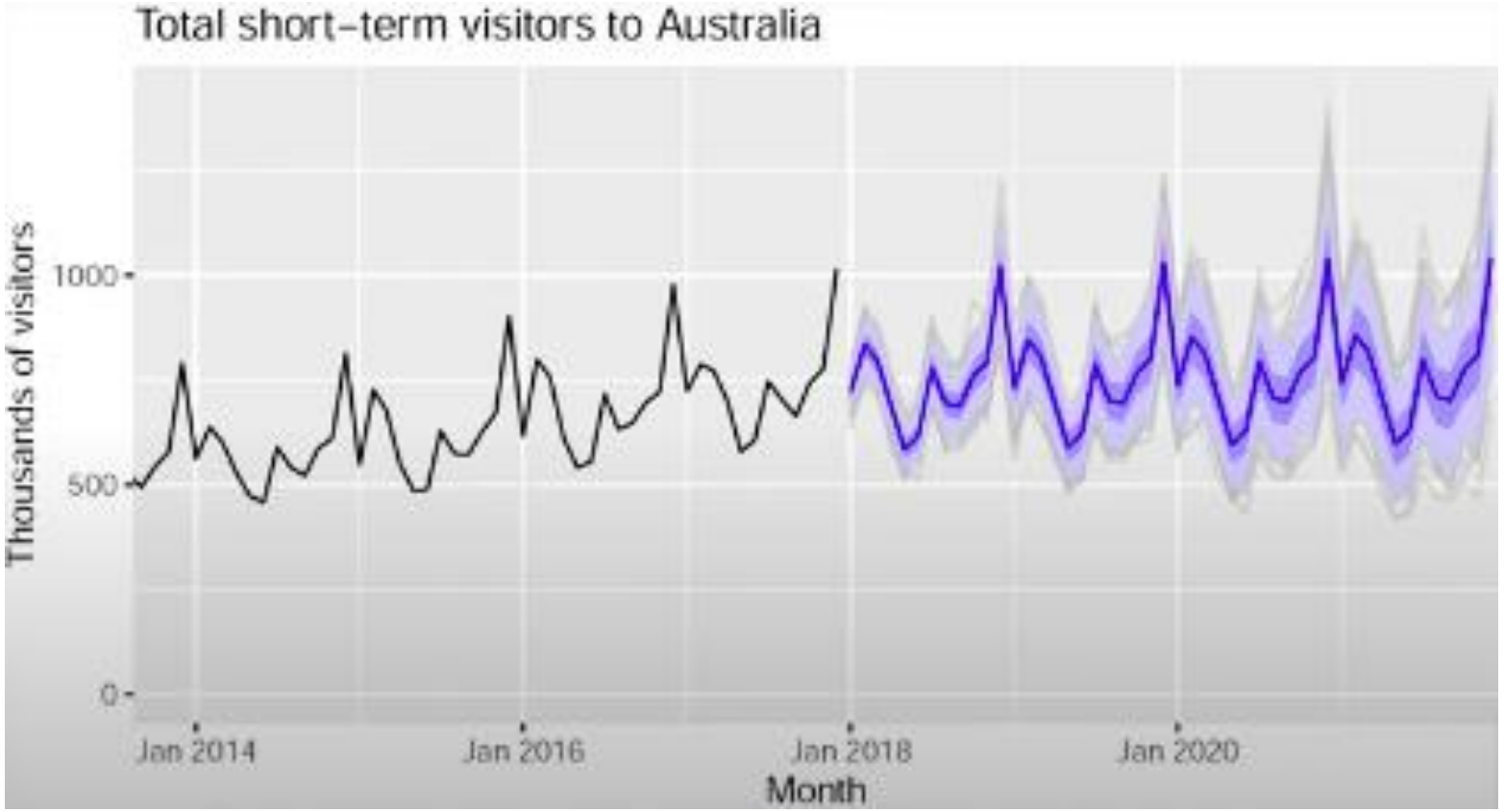
le nombre de visiteurs en australie. [2000, 2018]

Time series forecasting



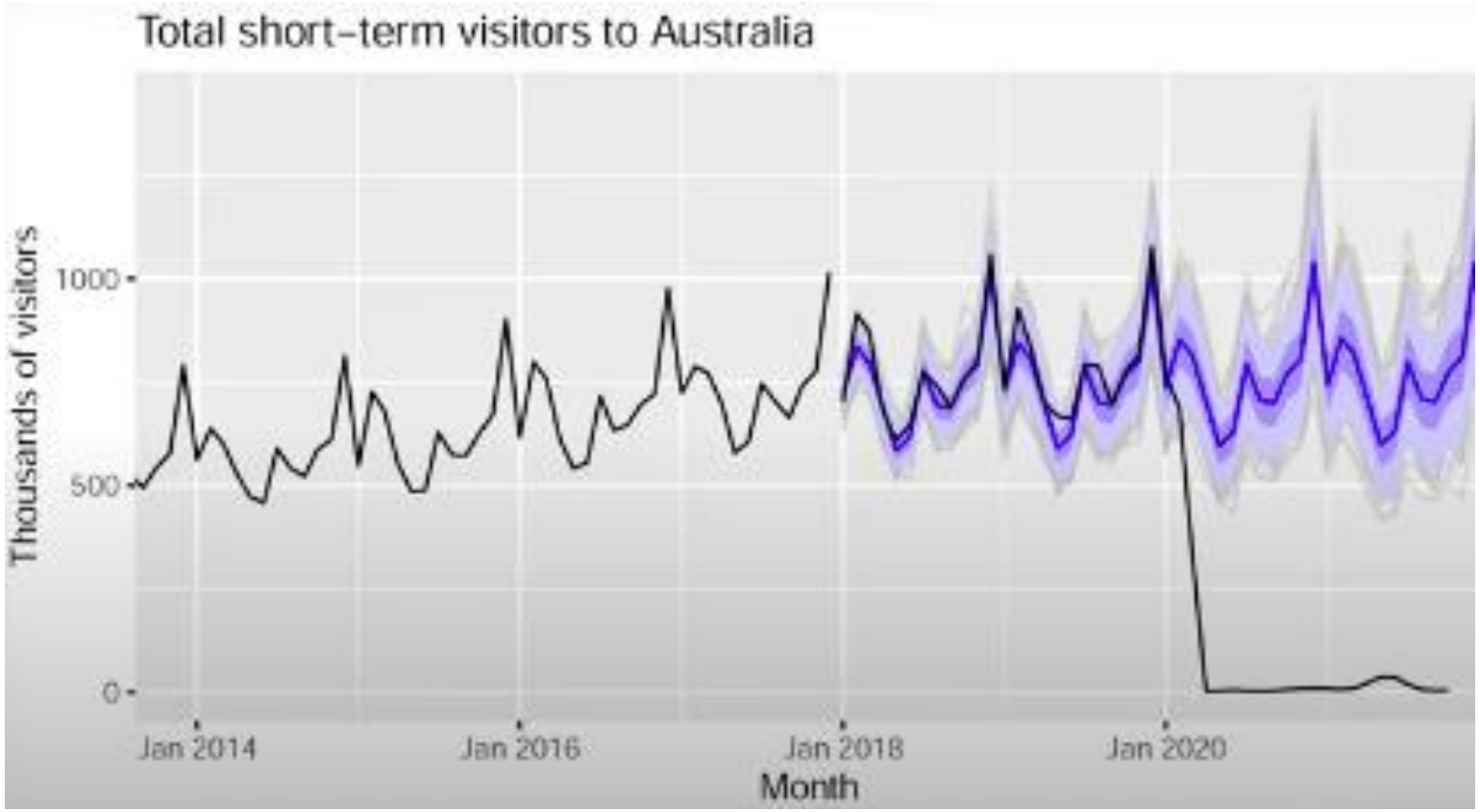
le nombre de visiteurs en australie. [2000, 2018]

Time series forecasting



le nombre de visiteurs en australie. [2000, 2018]

Time series forecasting



le nombre de visiteurs en australie. [2000, 2018]

Les motifs des séries temporelles (Time Series Patterns)

*" un bon modèle de prédiction du futur (forcasting) doit **capturer** tous les **motifs (patterns)** observés dans les données du passé."*

Quels sont les types des motifs qui caractérisent une Time Series?

Les motifs des séries temporelles (Time Series Patterns)

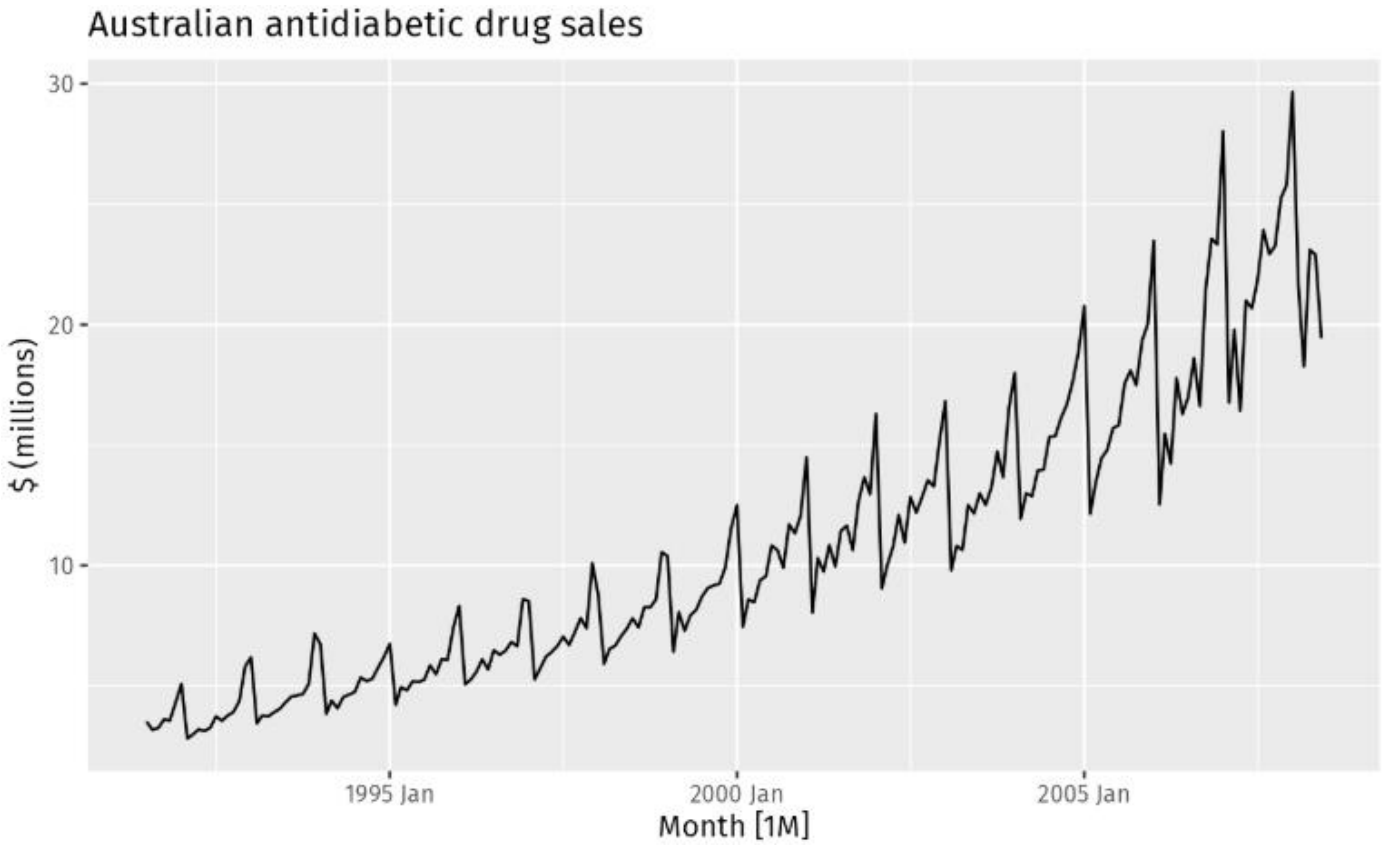
Les types des motifs qui caractérisent une Time Series:

- **Tendance (Trend)** :
  - Une augmentation ou une diminution progressive des valeurs au fil du temps.
- **Saisonnalité (Seasonality)** :
  - Un motif récurrent qui se répète à des intervalles fixes (mois, trimestre, année, etc.).
  - Exemple : L'augmentation des ventes de la glace chaque été.
- **Cycle:**
  - Un cycle se produit lorsque les données présentent des hausses et des baisses qui ne suivent pas une fréquence fixe.
- **Bruit (Noise)** :
  - Des fluctuations aléatoires qui n'ont pas de structure prévisible.

==> remarque: Si l'on élimine la **tendance** et la **saisonnalité**, il ne reste que le **bruit**.

Tendance (trend) & saisonnalité

Selon la figure, interpréter l'évolution des ventes de médicaments antidiabétiques.



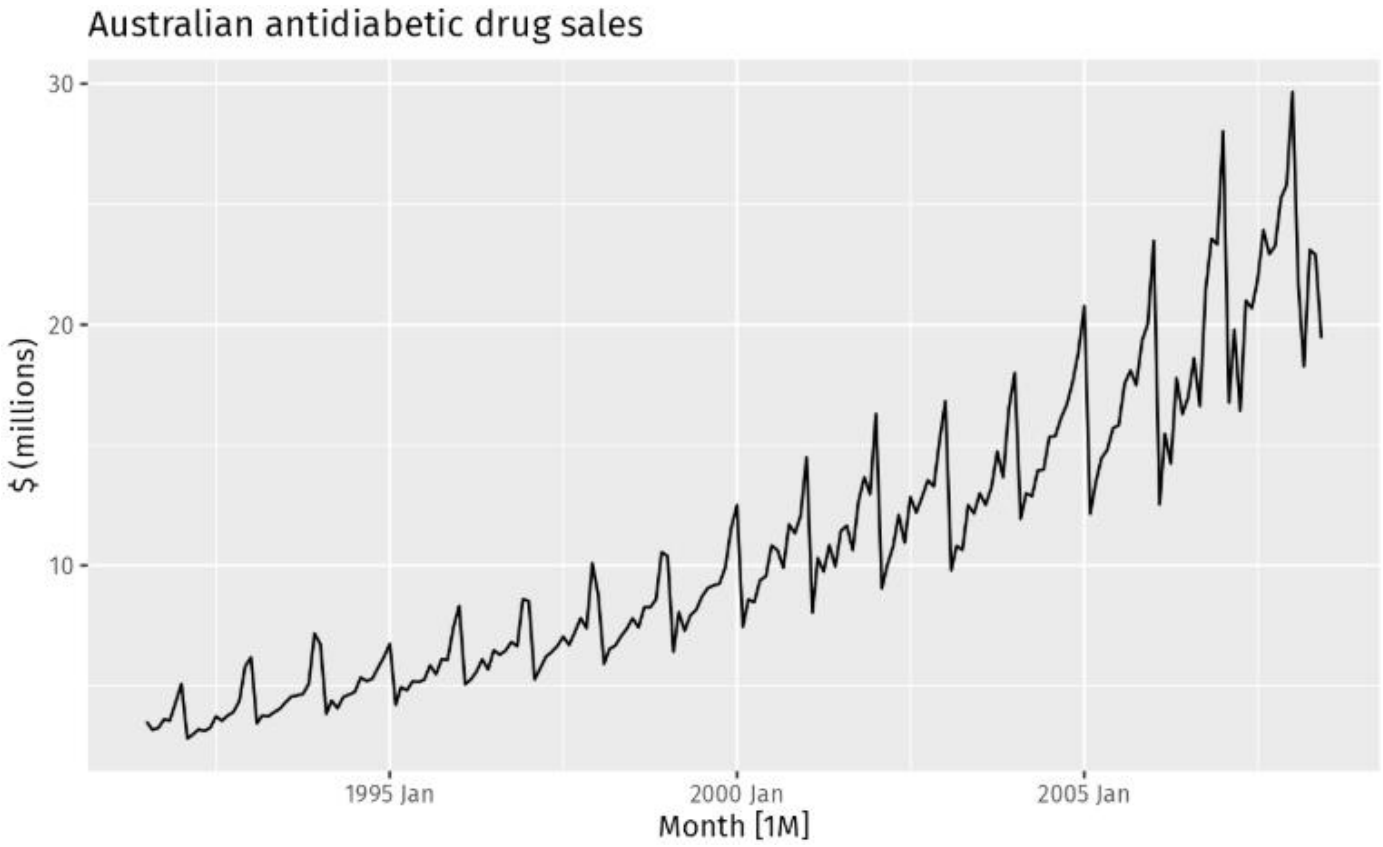
Les ventes mensuelles de médicaments antidiabétiques en Australie.

Tendance (trend)& saisonnalité

Interprétation de la figure:

- Tendance claire et croissante.
- Fort pattern saisonnier dont l'amplitude augmente lorsque la série s'élève.
- Chute soudaine au début de chaque année: causée par un programme de subvention gouvernemental qui incite les patients à faire des stocks de médicaments à la fin de l'année.

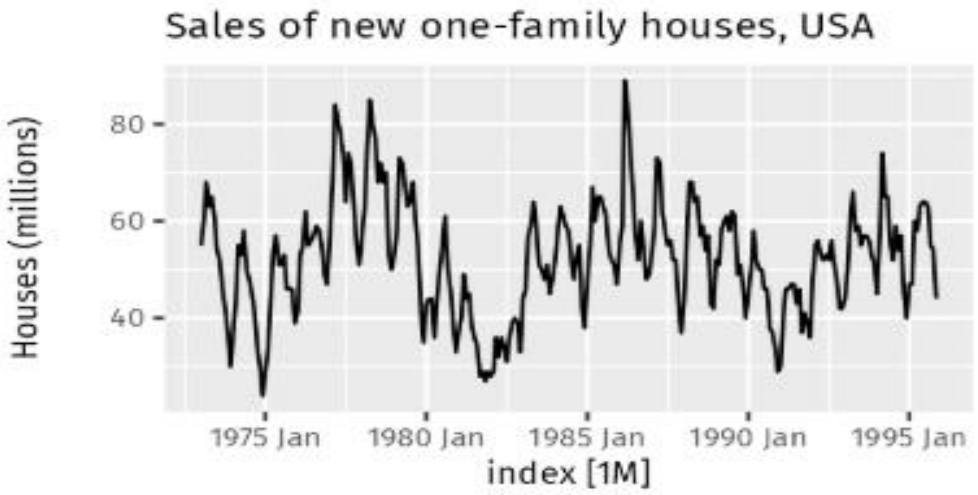
==> Toute prévision de cette série devrait capturer le pattern saisonnier ainsi que le fait que la tendance évolue lentement.

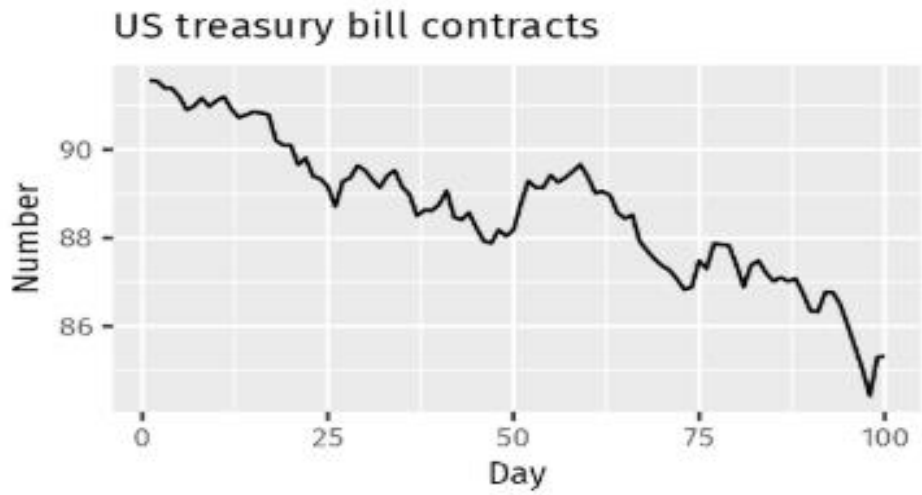


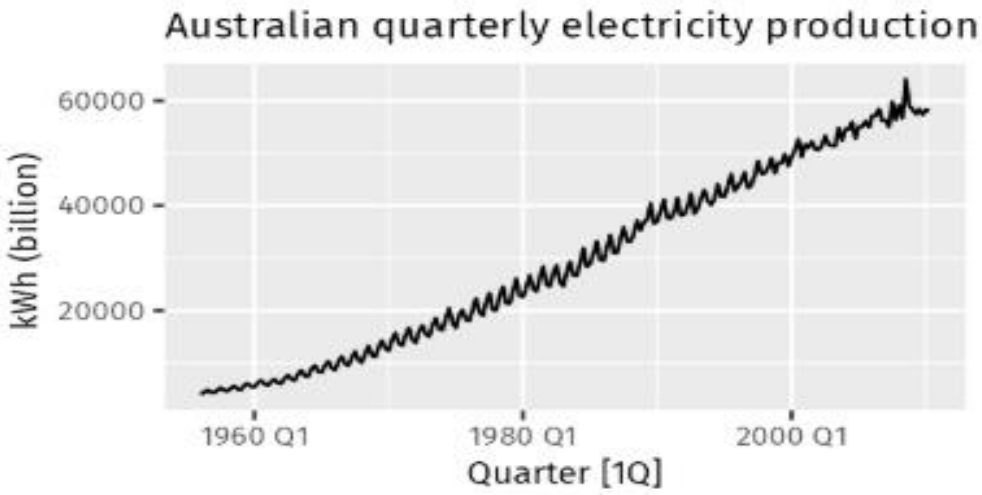
Les ventes mensuelles de médicaments antidiabétiques en Australie.

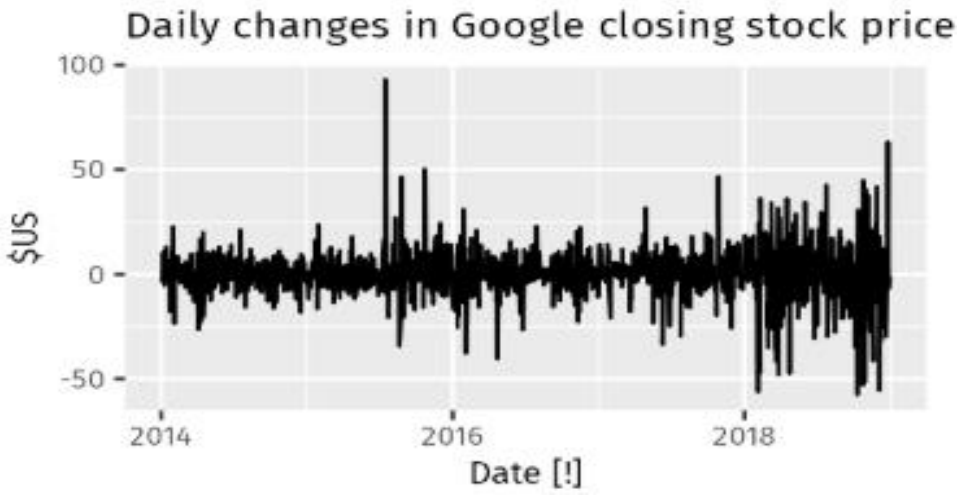
Exemples:

Interpréter les figures ci-après









Exemples:

- Les ventes mensuelles de logements (**en haut à gauche**) présentent une forte saisonnalité au sein de chaque année, ainsi qu'un comportement cyclique marqué avec une période d'environ 6 à 10 ans. Aucune tendance apparente ne se dégage des données sur cette période.

- Les contrats de bons du Trésor américain (**en haut à droite**) montrent les résultats du marché de Chicago sur 100 jours de trading consécutifs en 1981. Ici, il n'y a pas de saisonnalité, mais une tendance baissière évidente. Il est possible que, sur une période beaucoup plus longue, cette tendance à la baisse fasse en réalité partie d'un cycle plus large. Cependant, lorsqu'on observe seulement ces 100 jours, elle apparaît simplement comme une tendance.

- La production trimestrielle d'électricité en Australie (**en bas à gauche**) présente une tendance haussière marquée, ainsi qu'une forte saisonnalité. Il n'y a cependant aucune indication de comportement cyclique.

- La variation quotidienne du cours de clôture de l'action Google (**en bas à droite**) ne présente ni tendance, ni saisonnalité, ni comportement cyclique. Les fluctuations sont aléatoires et ne semblent pas prévisibles, sans motifs clairs pouvant être exploités pour le développement d’un modèle de prévision.

Les motifs des séries temporelles (Time Séries Patterns)

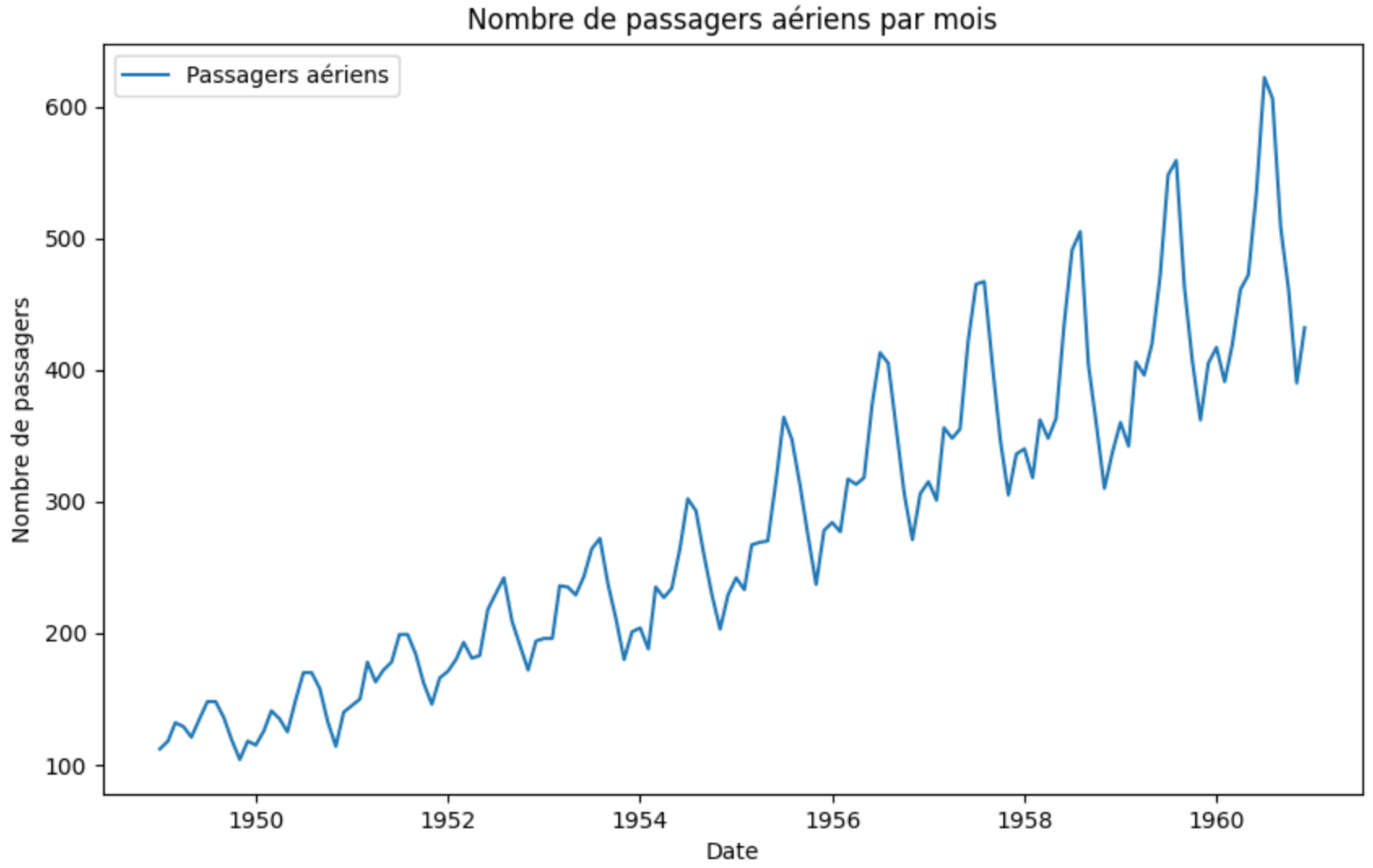
- **Analyse Exploratoire des séries temporelles:**

Les étapes à suivre pour analyser une série temporelle:

1. Visualiser la série temporelle et interpréter la figure.
2. Décomposition de la série temporelle (Trend, saisonnalité, résidus).
3. Étudier la Stationnarité (visuellement où Utiliser le graphique ACF (autocorrélation))

Exemples d'analyse Exploratoire

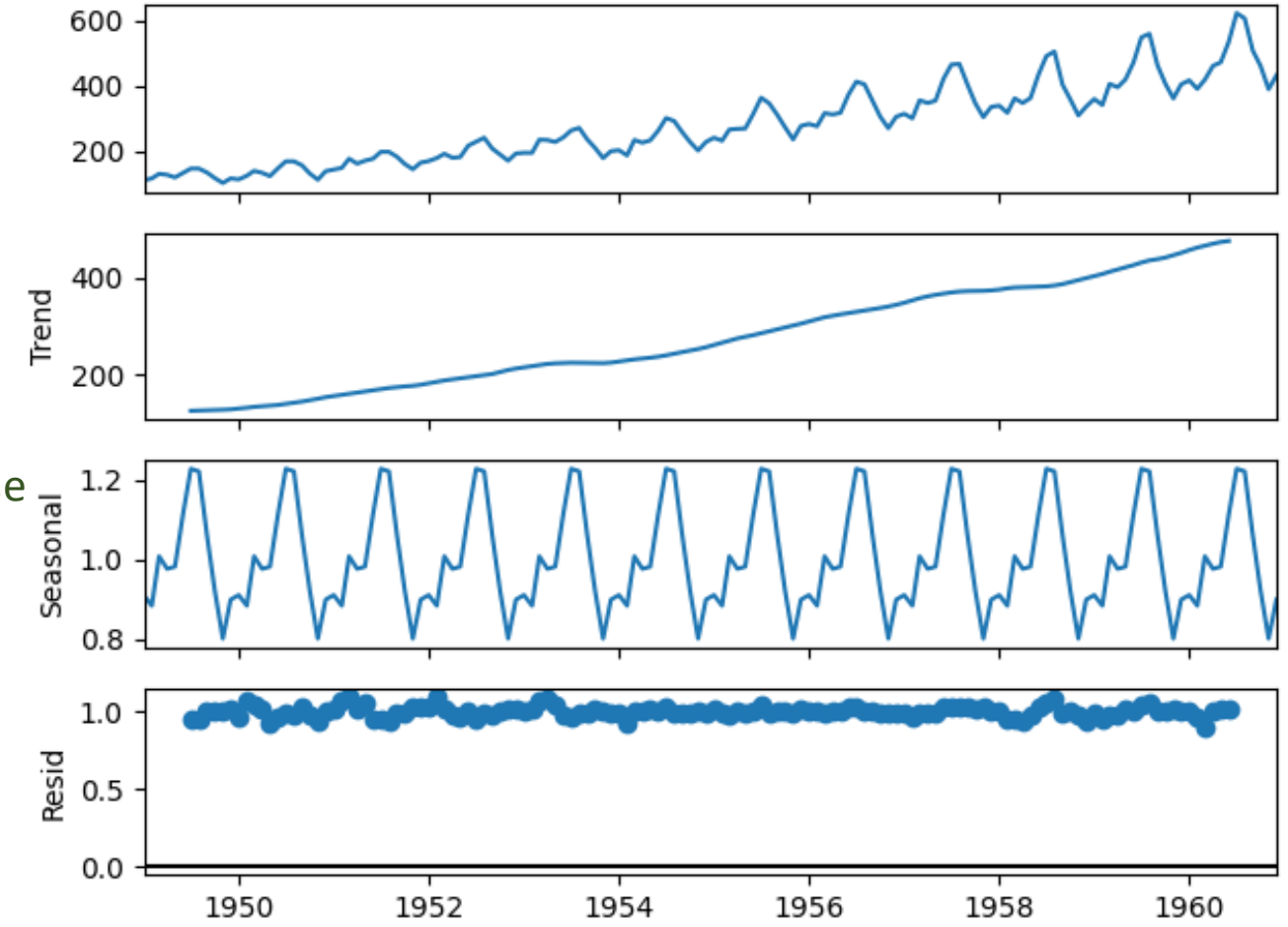
1. Visualiser la série temporelle et interpréter la figure.



Exemples d'analyse Exploratoire

2. Décomposition de la série temporelle:
   - **Trend**
   - **Saisonnalité**
   - **résidus**

```
from statsmodels.tsa.seasonal import seasonal_decompose

sd = seasonal_decompose(data, model='multiplicative')

sd.plot()

plt.show()
```



Exemples d'analyse Exploratoire

3. Étudier la **Stationnarité**

- **Qu'est-ce que la stationnarité ?**

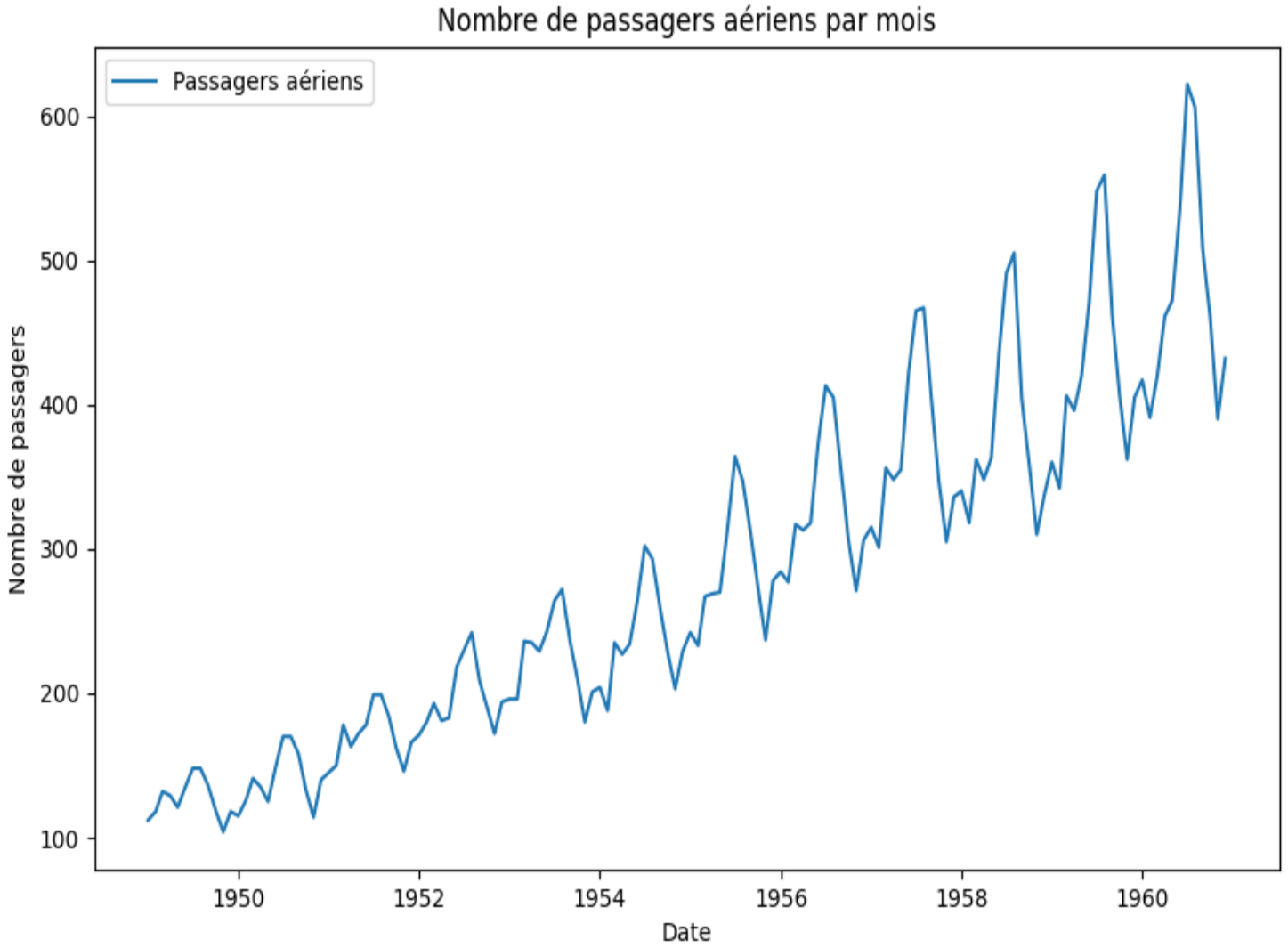
Une série temporelle est dite stationnaire si ses propriétés statistiques (moyenne, variance, autocorrélation) restent constantes dans le temps. Concrètement, cela signifie :

- **La moyenne** de la série **ne change pas** au fil du temps.
- **La variance est constante** (pas d'hétéroscédasticité).
- **L'autocorrélation** entre les observations est faible .

Exemples d'analyse Exploratoire

3. Étudier la **Stationnarité:**

Est-ce que cette série temporelle est stationnaire?



Exemples d'analyse Exploratoire

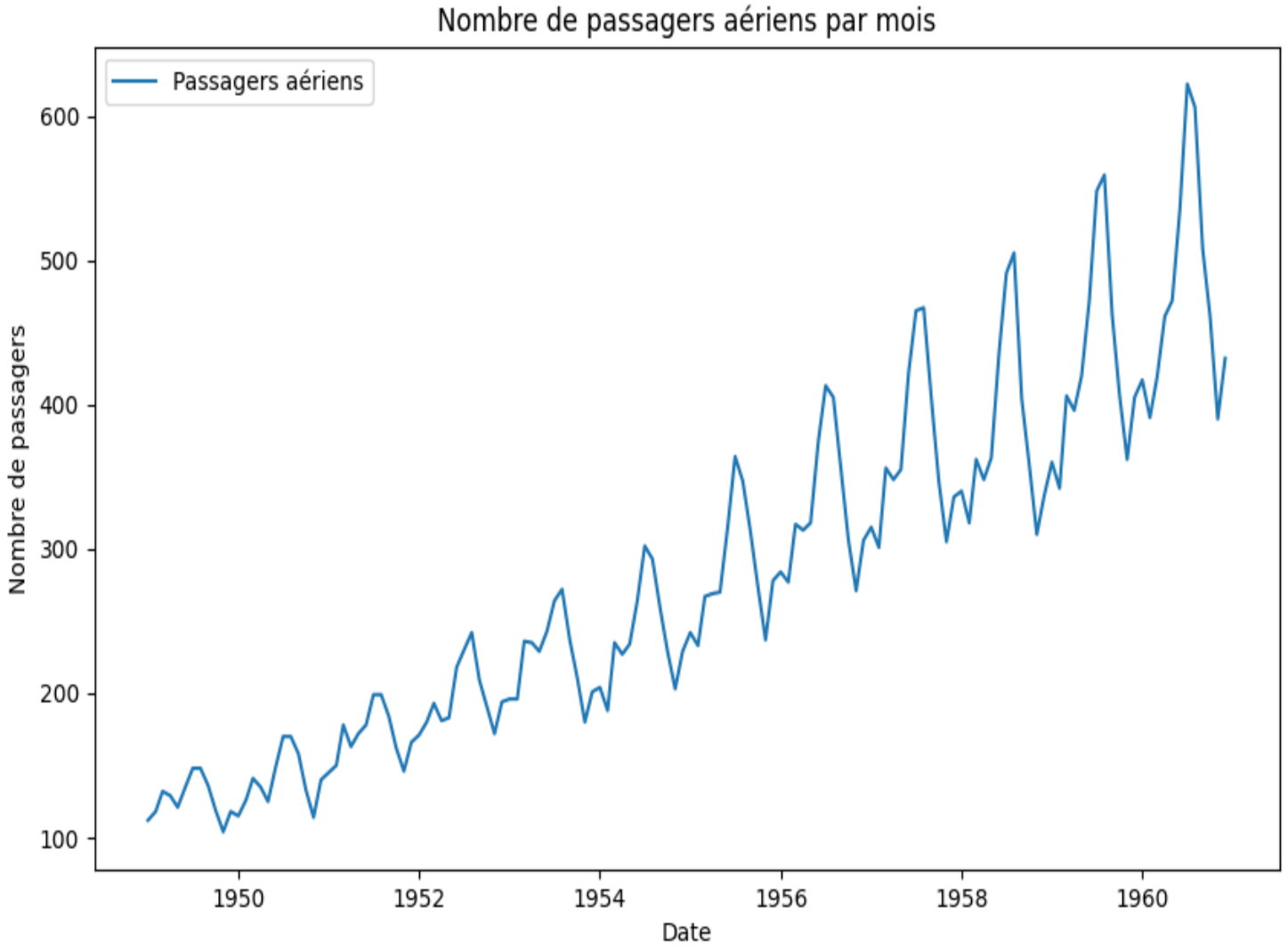
3. Étudier la **Stationnarité:**

Est-ce que cette série temporelle est stationnaire?

n'est pas stationnaire:

- Moyenne n'est pas stable.
- Variance n'est pas stable.

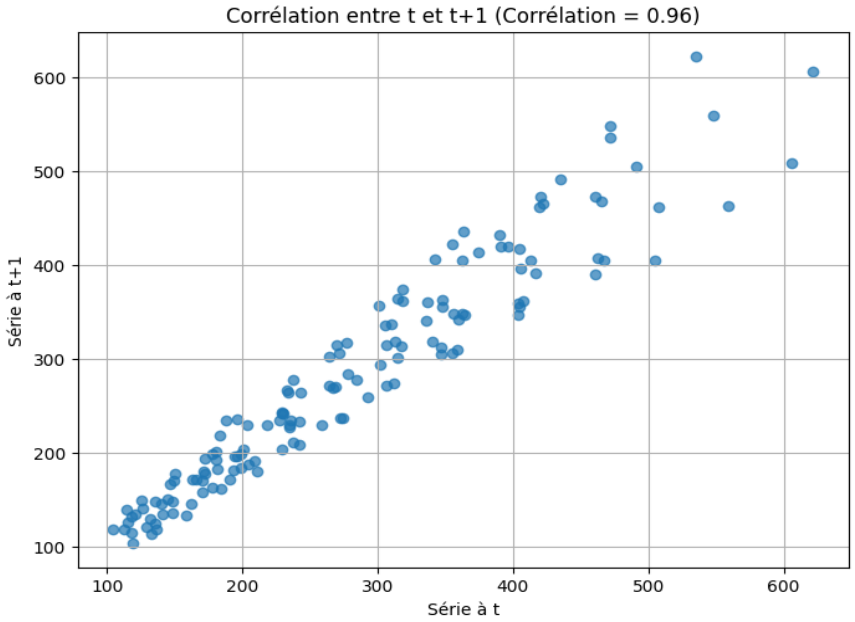
Comment mesurer l'autocorrélation?

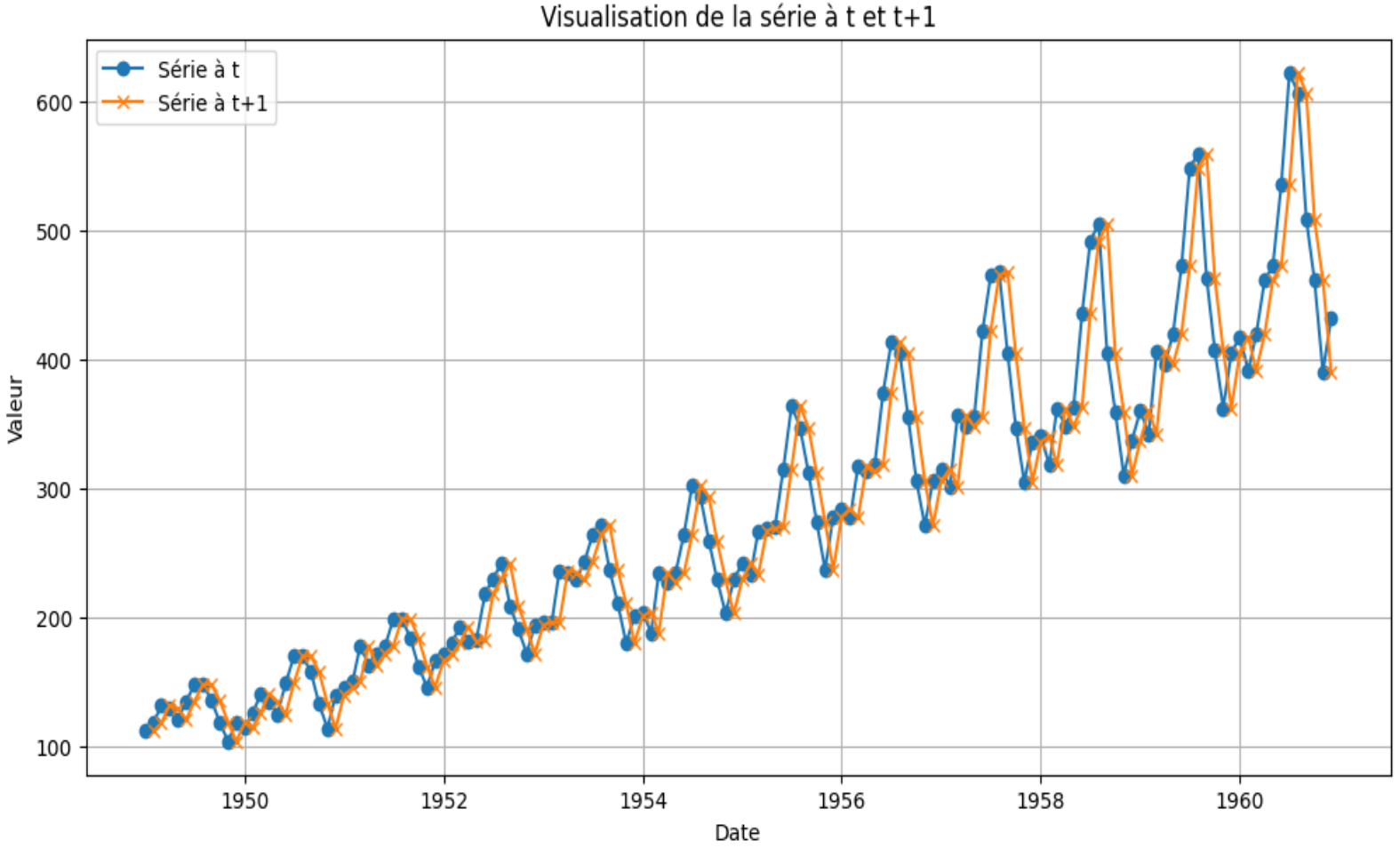


Exemples d'analyse Exploratoire

- **l'autocorrélation:**

la corrélation entre une série temporelle à un temps t et à un temps t+1 (c'est-à-dire avec un **décalage** de 1..n, ou **lag** = 1..n),

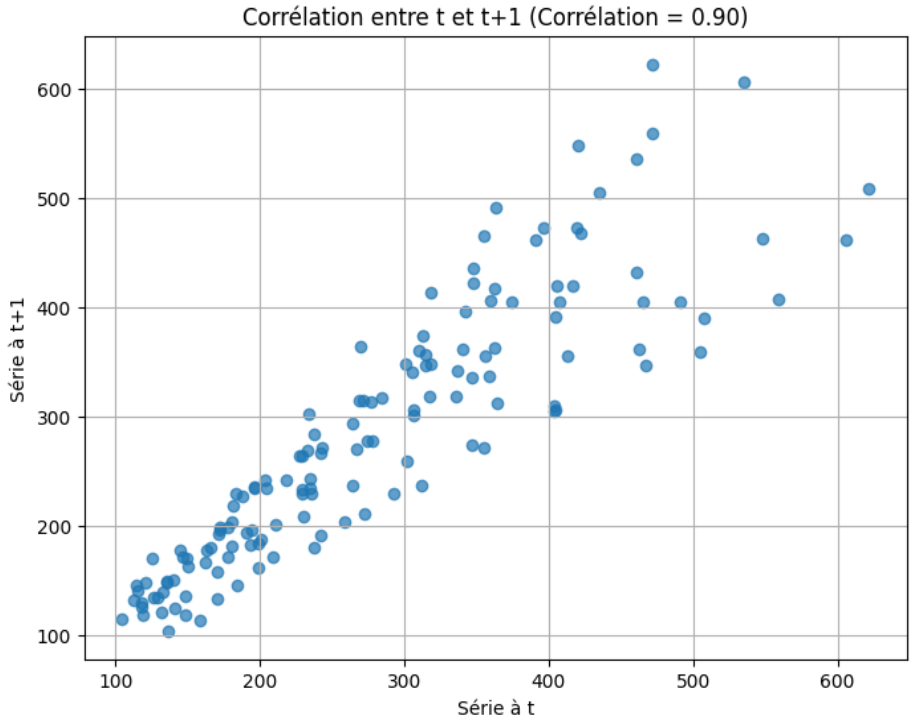


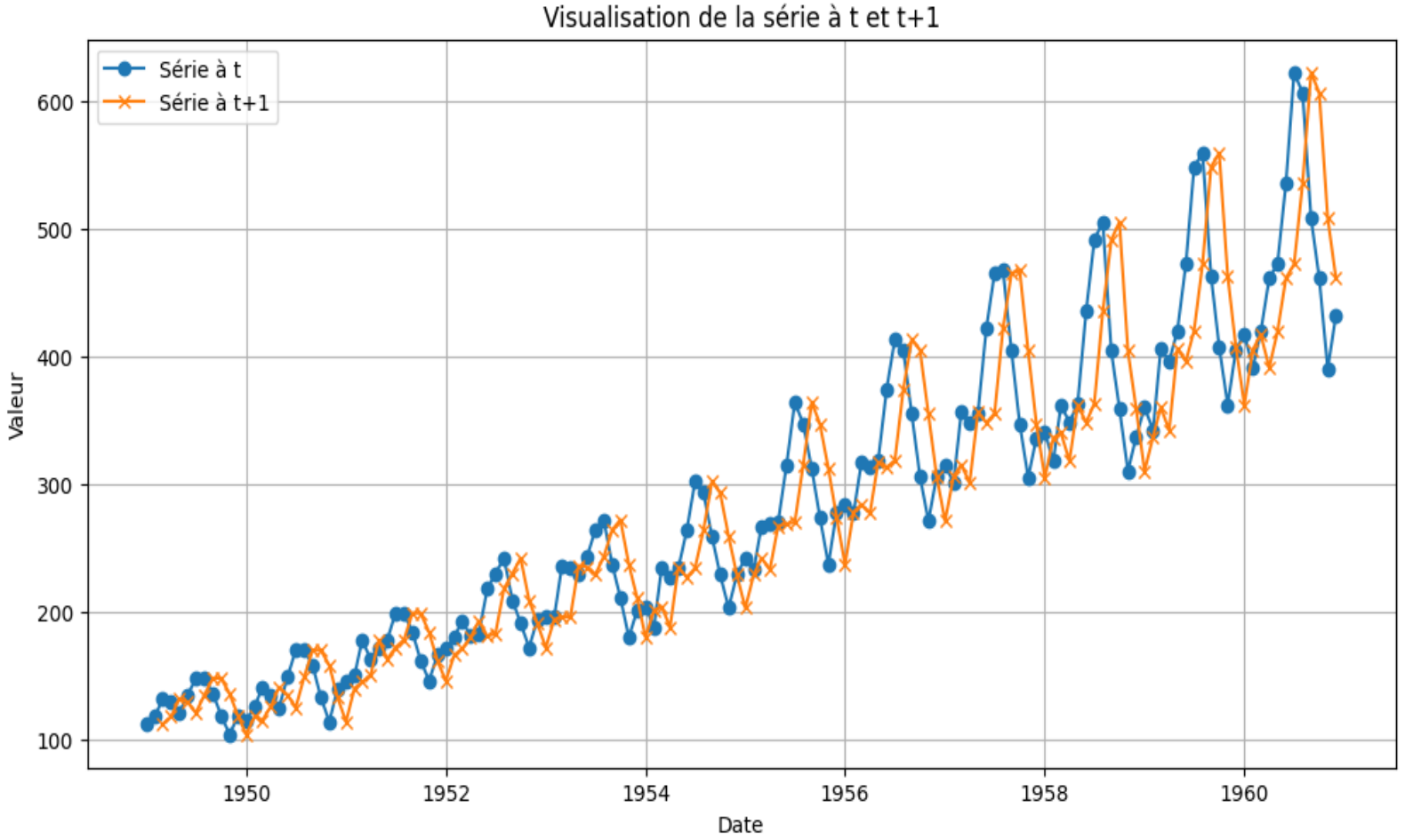


Lag = 1

Exemples d'analyse Exploratoire

- **l'autocorrélation:**

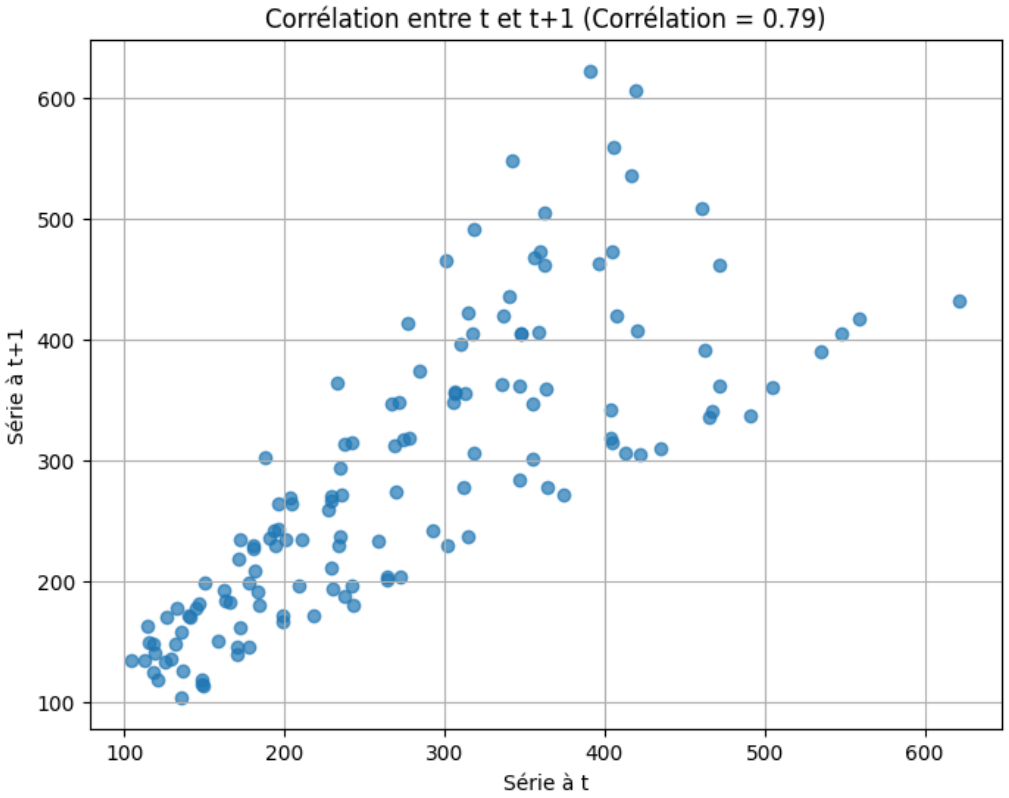


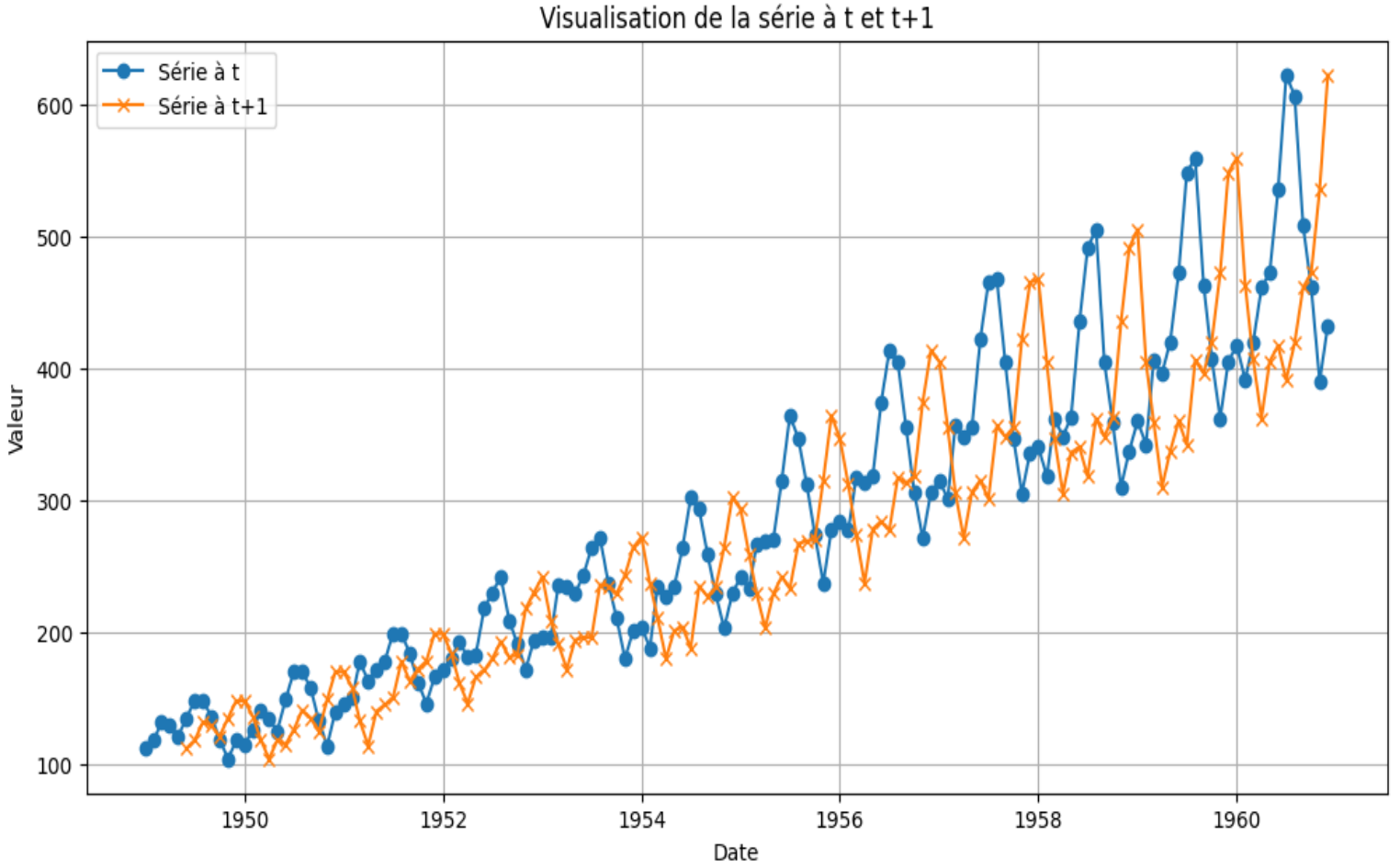


Lag = 2

Exemples d'analyse Exploratoire

- **l'autocorrélation:**

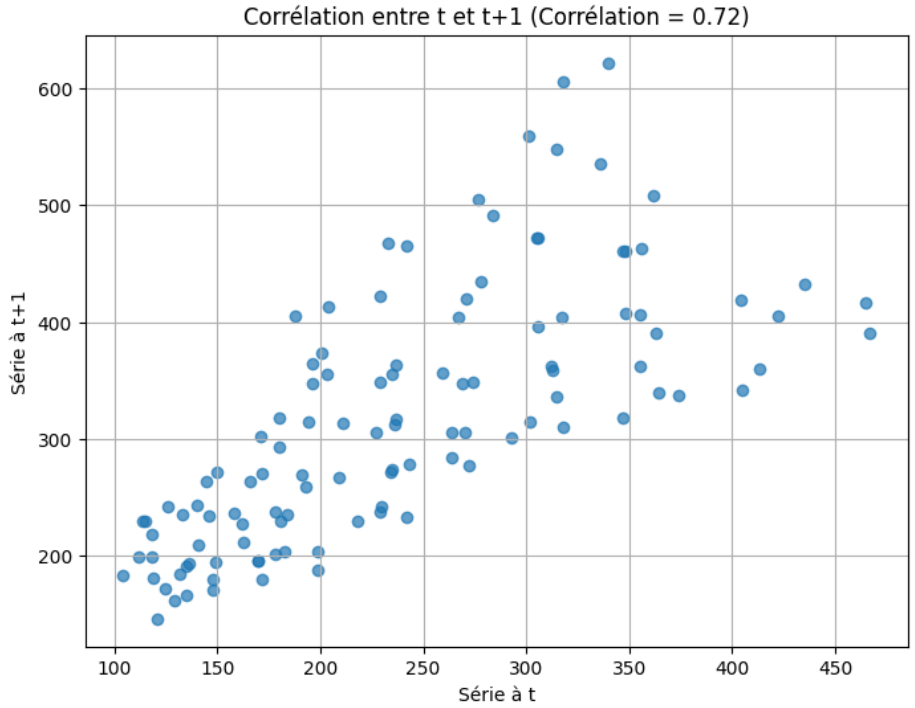


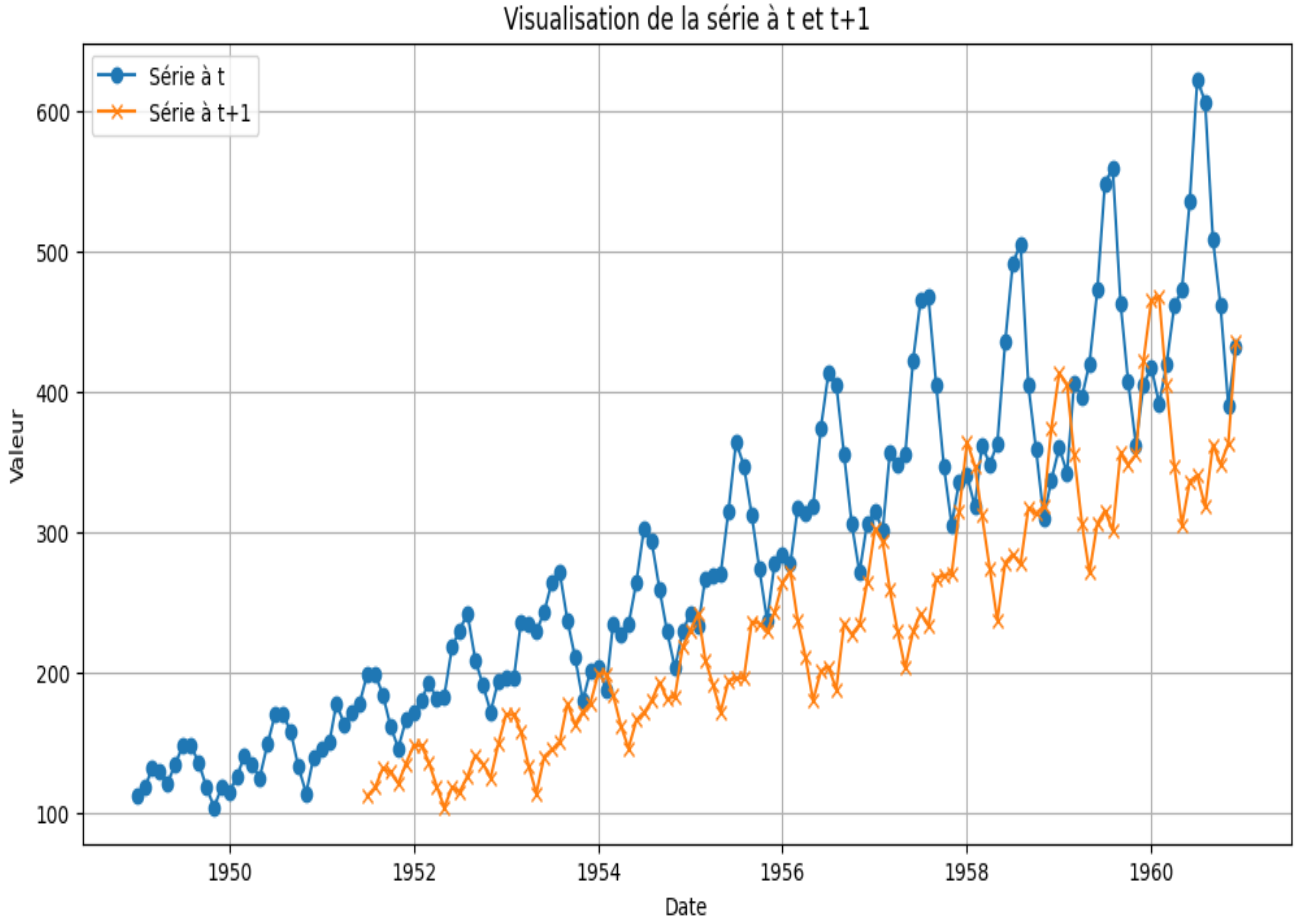


Lag = 5

Exemples d'analyse Exploratoire

- **l'autocorrélation:**

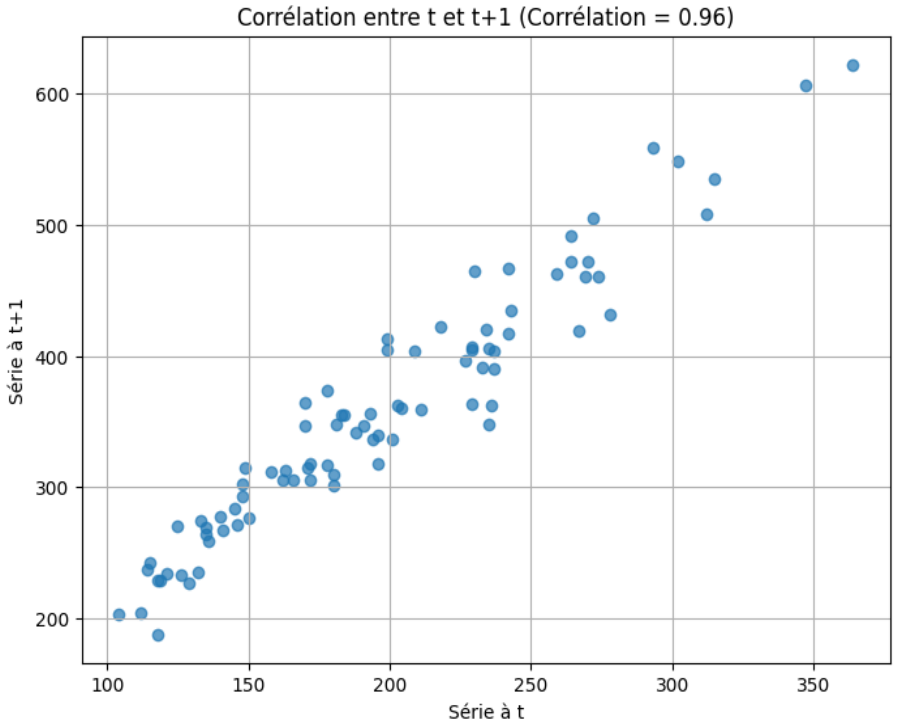


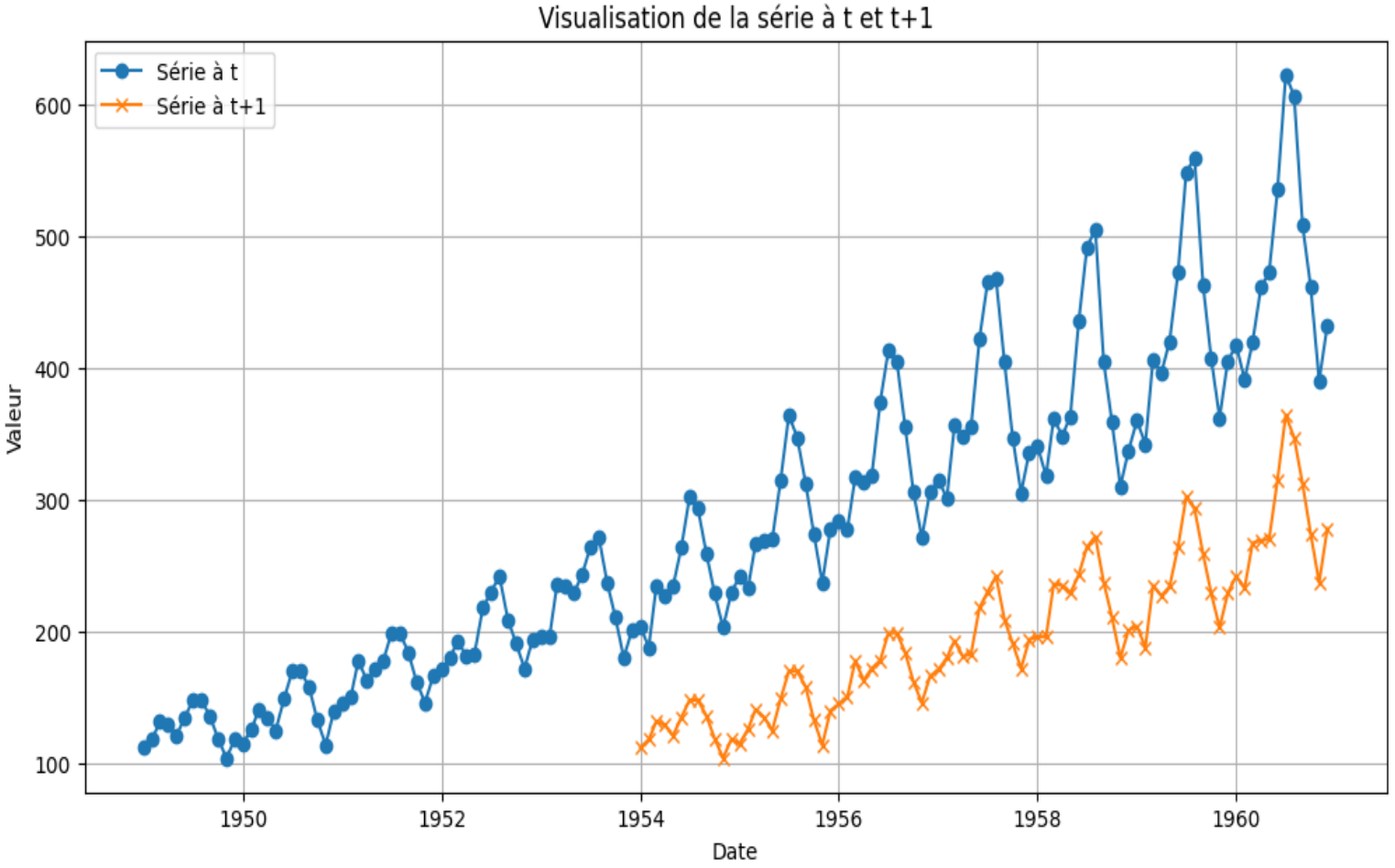


Lag = 30

Exemples d'analyse Exploratoire

- **l'autocorrélation:**





Qu'est-ce que vous remarquez?

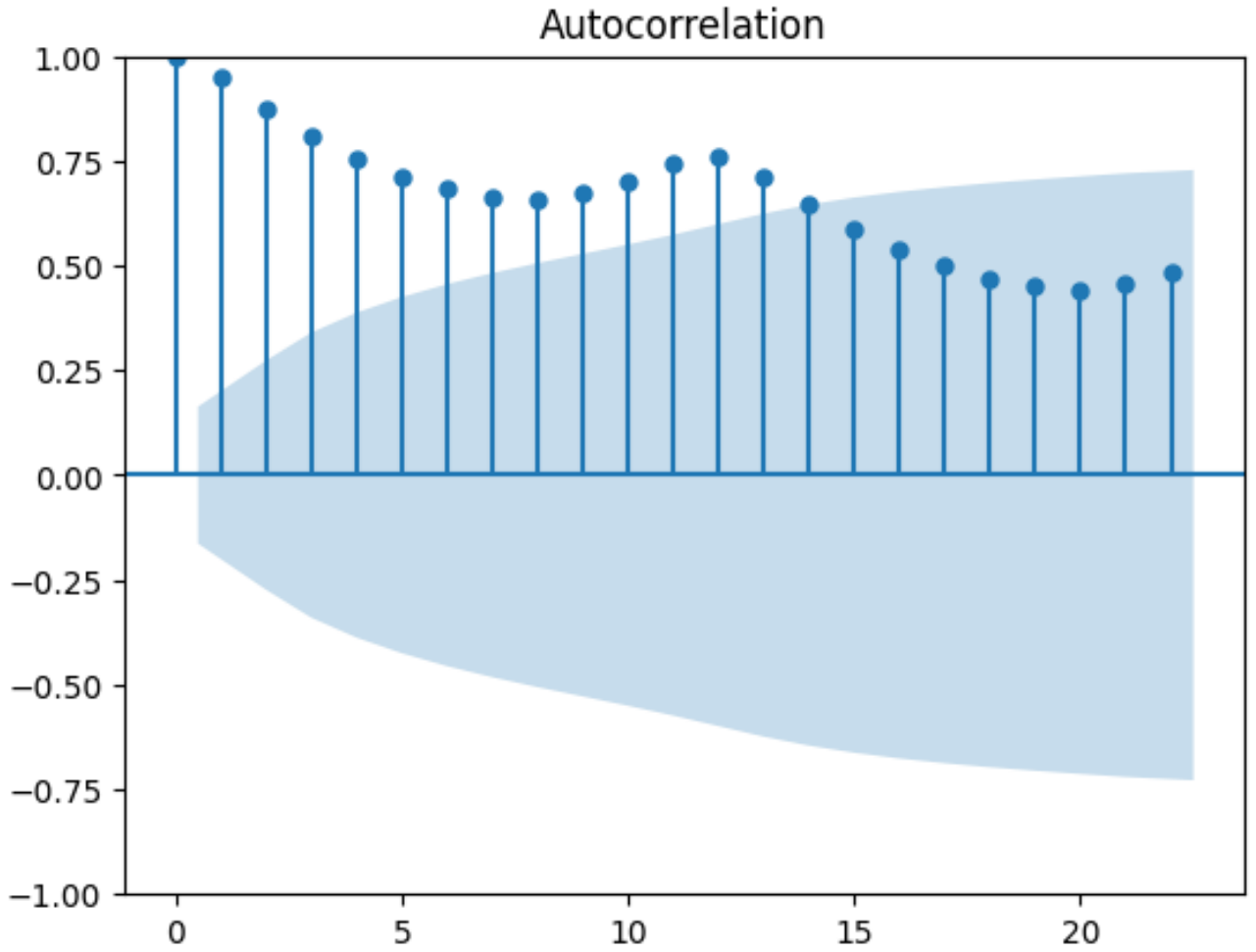
Lag = 60

Exemples d'analyse Exploratoire

- **l'autocorrélation:**

On peut résumer toutes les étapes avant par l'application de la méthode **ACF (AutoCorrelation Function)**

```
from statsmodels.graphics.tsaplots import plot_acf

plot_acf(data)

 plt.show()
```



L'autocorrélation entre les observations est **forte:**

- **Trend.**
- **Saisonnalité.**

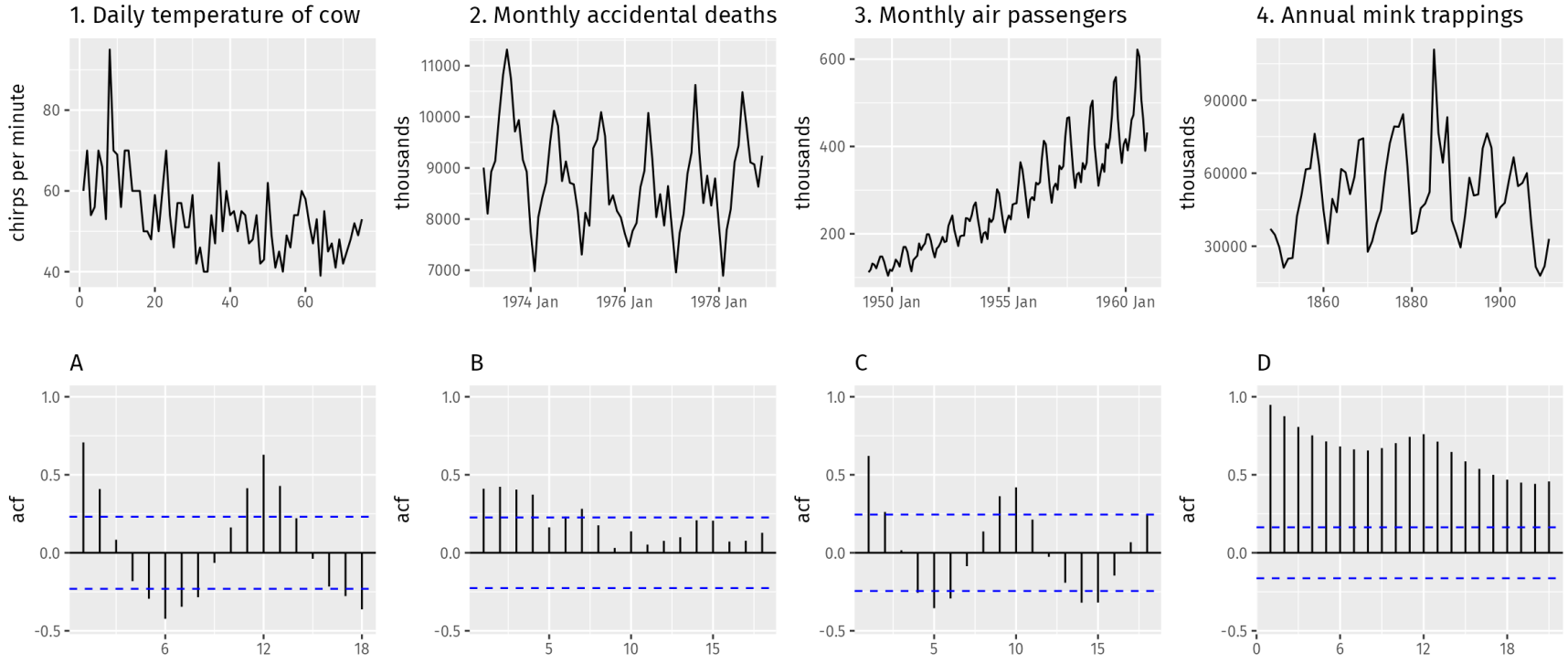
n'est pas stationnaire

Lag = 1-->25

Exemples d'analyse Exploratoire

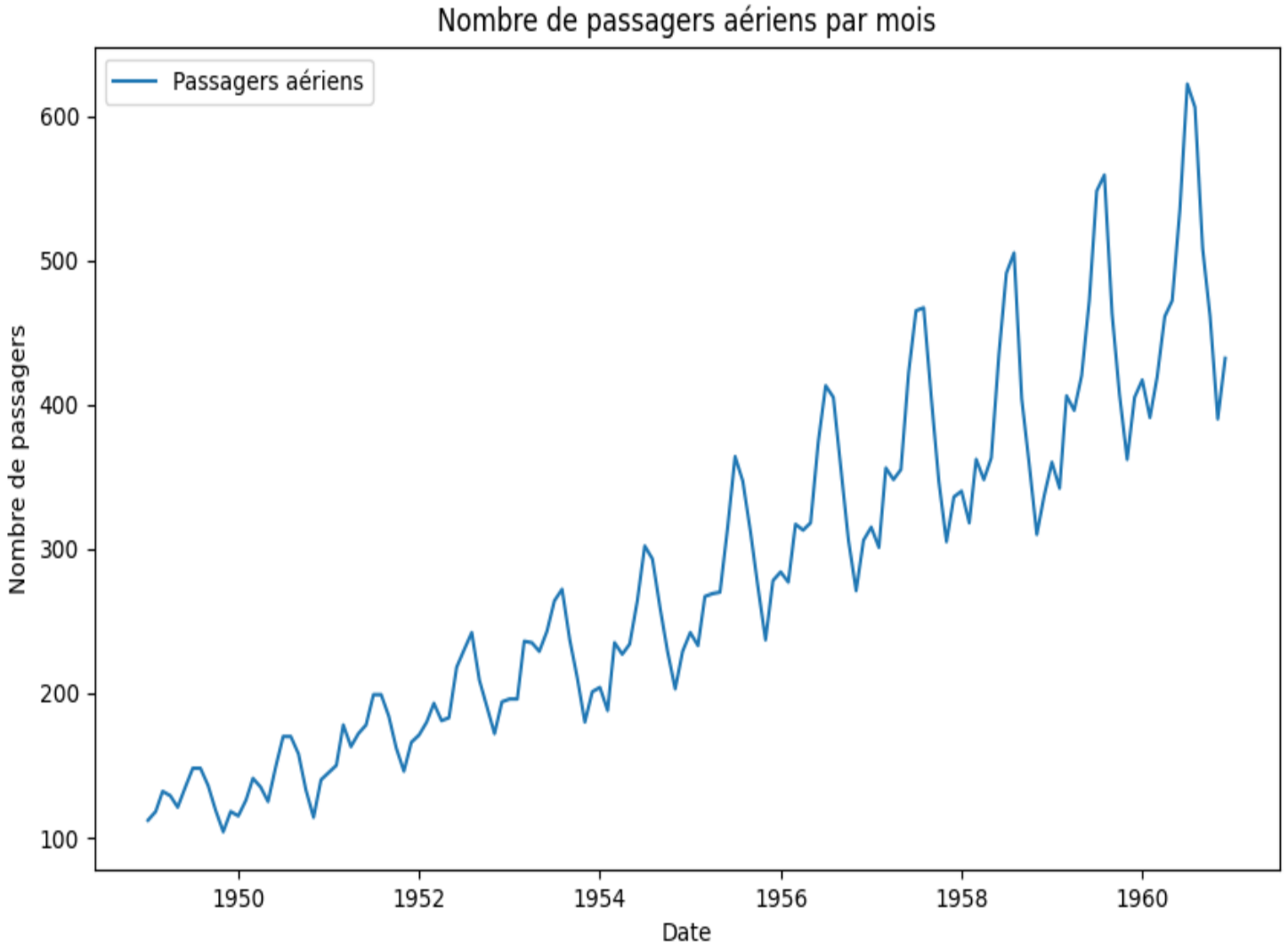
- **l'autocorrélation:**

Les graphiques temporels et les graphiques ACF suivants correspondent à quatre séries temporelles différentes. Votre tâche consiste à associer chaque graphique temporel de la première ligne à l'un des graphiques ACF de la deuxième ligne.



Exemples d'analyse Exploratoire

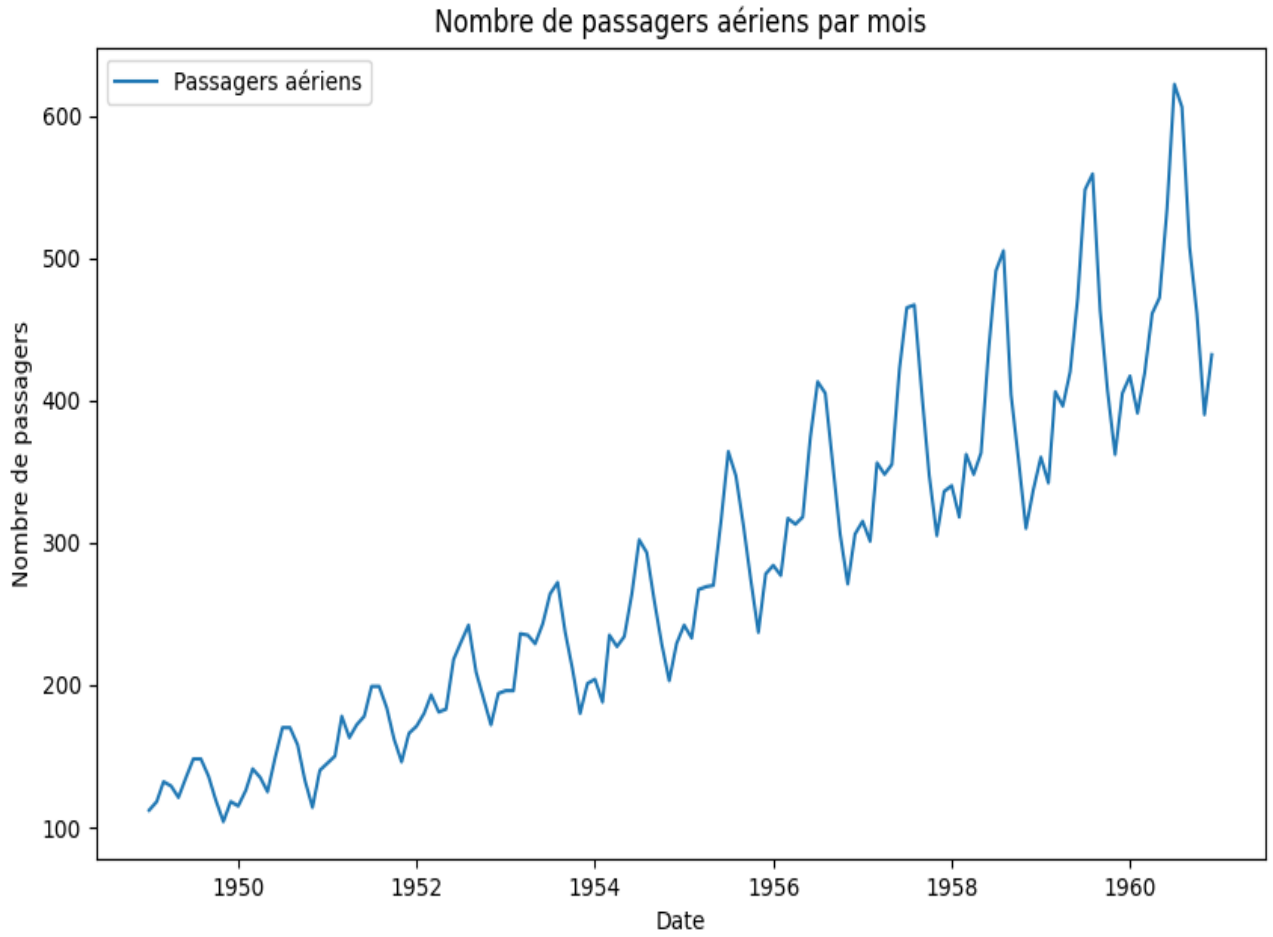
- comment éliminer la tendance dans une série temporelle?



Exemples d'analyse Exploratoire

- comment éliminer la tendance dans une série temporelle?

_ Appliquer une Différenciation:
La différenciation permet d'éliminer la tendance en soustrayant chaque valeur de la valeur précédente.



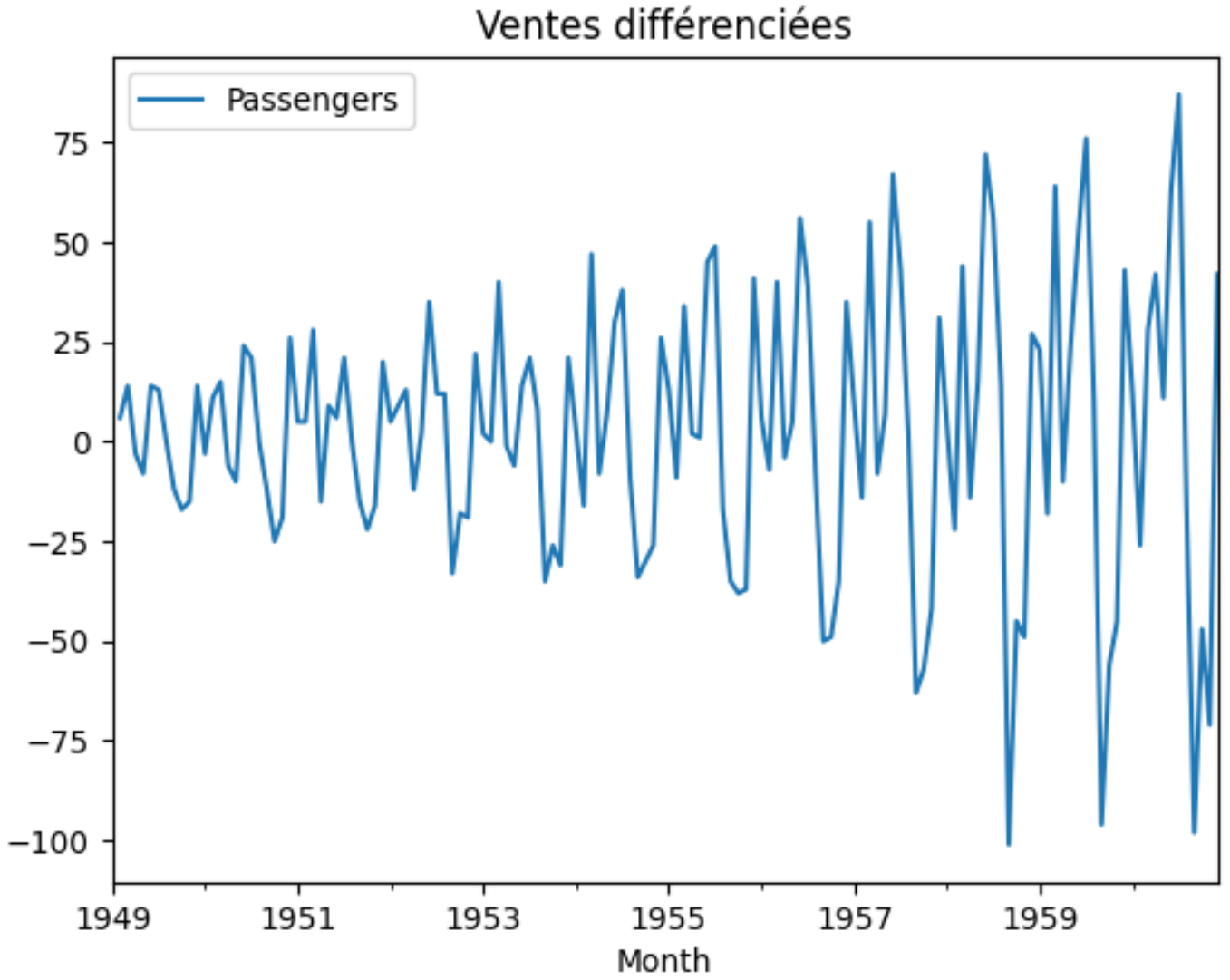
Exemples d'analyse Exploratoire

- comment éliminer la tendance dans une série temporelle?

_ Appliquer une Différenciation:
La différenciation permet d'éliminer la tendance en soustrayant chaque valeur de la valeur précédente.

différenciation de l'ordre 1:

```
data_diff =data.diff(1)

# Visualiser la série différenciée
plt.figure(figsize=(18, 16))
data_diff.plot(title='Ventes différenciées')
plt.show()
```



==>est ce que la série devient **stationnaire**?il faut appliquer le test statistique **adfuller**

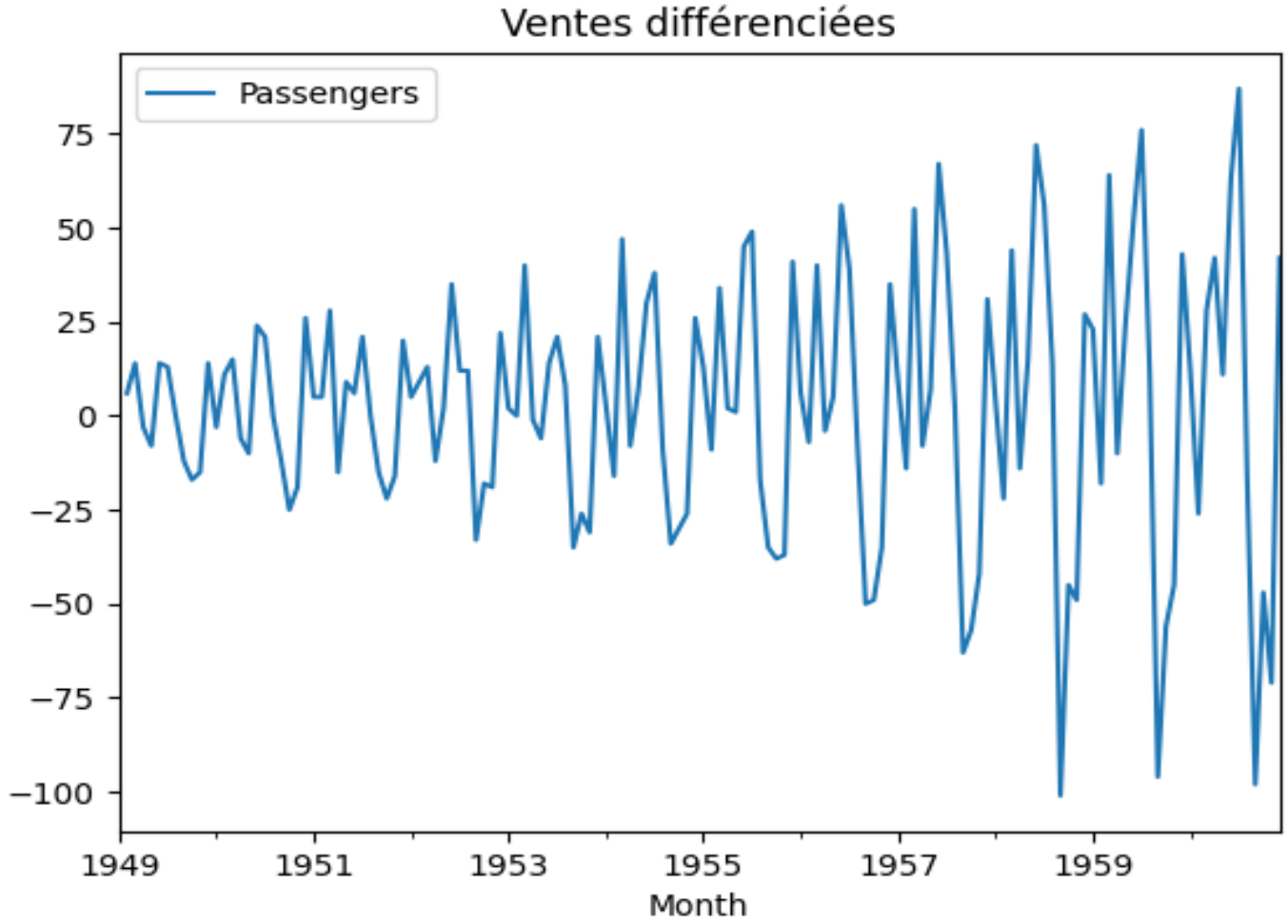
Exemples d'analyse Exploratoire

```
from statsmodels.tsa.stattools import adfuller

result = adfuller(data_diff.dropna())

print("Statistique du test ADF :", result[0])
print("p-value :", result[1])
print("Valeurs critiques :", result[4])

if result[1] > 0.05:
        print("La série n'est pas stationnaire.")
else:
        print("La série est stationnaire.")
```

==>est ce que la série devient **stationnaire**?

La série n'est pas stationnaire

Il faut passer à l'ordre suivant de différenciation



Exemples d'analyse Exploratoire

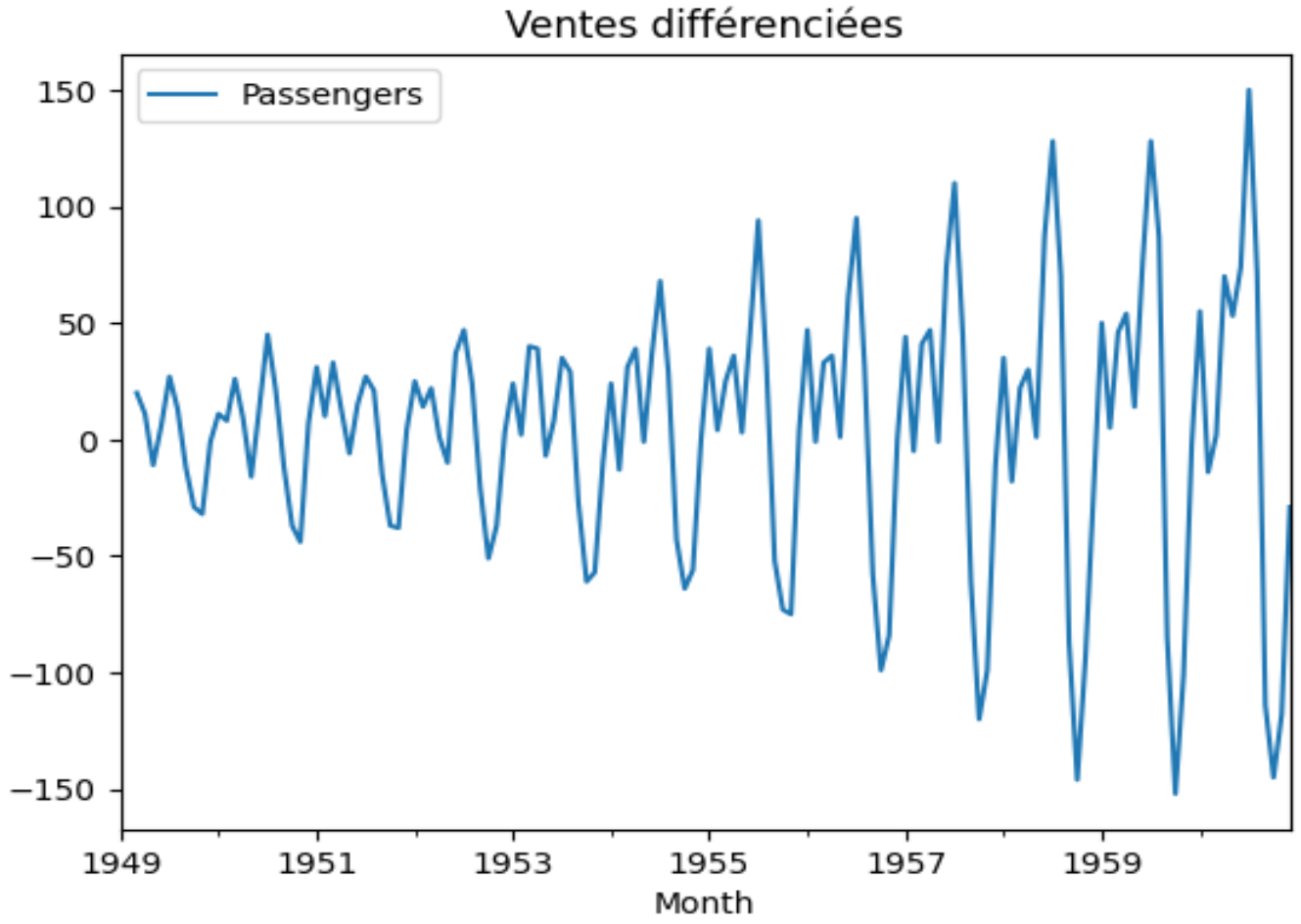
Différenciation de l'ordre 2:

```
data_diff =data.diff(2)

# Visualiser la série différenciée
plt.figure(figsize=(18, 16))
data_diff.plot(title='Ventes différenciées')
plt.show()
```

test statistique adfuller:

La série est stationnaire.

==>est ce que la **variance** est constante?



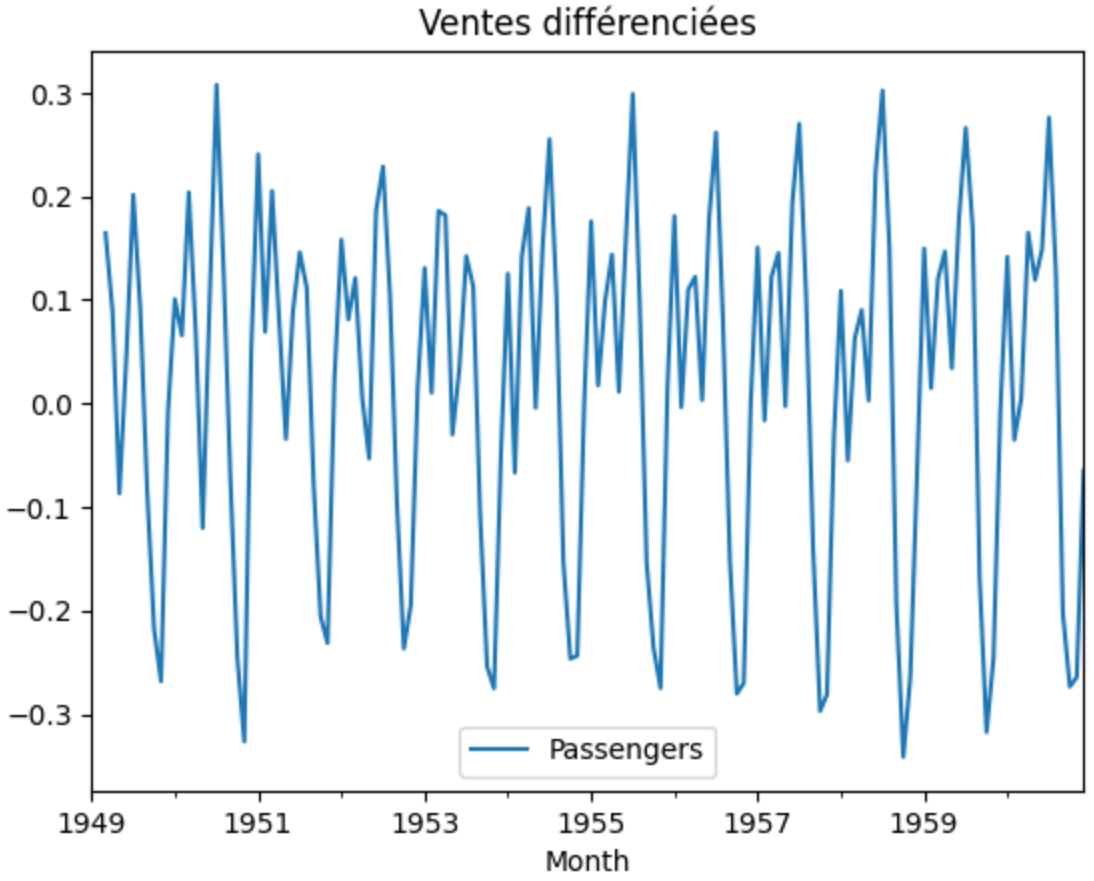
Exemples d'analyse Exploratoire

==>est ce que la **variance** est constante?**non**

il faut appliquer une transformation logarithmique:

ts_log = np.log(data)

il faut appliquer une transformation logarithmique avant d'appliquer la différenciation



Utiliser un modèle de prévision (forcasting)

- La stationnarité est essentielle en analyse des séries temporelles, notamment pour utiliser des modèles comme **ARIMA**, qui supposent que la série est stationnaire pour faire des prévisions fiables.
  - **ARIMA** est adapté aux séries temporelles non stationnaires sans saisonnalité.
  - **SARIMA** (Saisonnière ARIMA) prend en compte la saisonnalité en ajoutant des composantes saisonnières au modèle ARIMA.

Utiliser un modèle de prévision (forcasting)

- La stationnarité est essentielle en analyse des séries temporelles, notamment pour utiliser des modèles comme **ARIMA**, qui supposent que la série est stationnaire pour faire des prévisions fiables.
  - **ARIMA** est adapté aux séries temporelles non stationnaires sans saisonnalité.
  - **SARIMA** (Saisonnière ARIMA) prend en compte la saisonnalité en ajoutant des composantes saisonnières au modèle ARIMA.

Utiliser un modèle de prévision (forcasting)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from pmdarima import auto_arima
# Trouver le meilleur modèle SARIMA
model = auto_arima(data["Passengers"], seasonal=True, m=12, trace=True, stepwise=True)
# Afficher les paramètres optimaux
print(model.summary())
# Ajuster le modèle SARIMA optimal
sarima_model = sm.tsa.statespace.SARIMAX(data["Passengers"], order=model.order,
seasonal_order=model.seasonal_order)
sarima_result = sarima_model.fit()
# Prévision: methode 1
Prédiction data["Prévision"] = sarima_result.predict(start=len(data), end=len(data)+24, dynamic=True)
```

Utiliser un modèle de prévision (forcasting)

```
# Prévision: methode 2
n_periods = 24  # Nombre de mois à prédire
forecast = sarima_result.get_forecast(steps=n_periods)
predictions = forecast.predicted_mean
```

```
1961-01-01    445.634934
1961-02-01    420.395024
1961-03-01    449.198348
1961-04-01    491.839976
1961-05-01    503.394502
1961-06-01    566.862471
1961-07-01    654.260189
1961-08-01    638.597493
1961-09-01    540.883739
1961-10-01    494.126610
```

```
# Intervalle de confiance
conf_int = forecast.conf_int()
```

	lower Passengers	upper Passengers
1961-01-01	423.344848	467.925020
1961-02-01	394.235397	446.554652
1961-03-01	419.831711	478.564985
1961-04-01	460.376259	523.303693
1961-05-01	470.380453	536.408552
1961-06-01	532.699319	601.025622
1961-07-01	619.221129	689.299249

Utiliser un modèle de prévision (forcasting)