



# DATA SCIENCE AVEC PYTHON

Année académique 2024-2025

Master: X-MAS-DE-4

Enseignant : David Rhenals

**Clause de confidentialité :** Ce cours est à usage unique des étudiants de l'EFREI, la diffusion externe de tout contenu de ce cours est strictement interdite sans l'autorisation écrite préalable de l'enseignant.

# Analyse de données structurées avec la librairie Pandas

# Contenu

- Qu'est-ce que Pandas?
- Structures de données gérées par pandas (Series et Data Frames) ?
- Principales attributs des séries et des Data frames en pandas
- Indexation et découpage en pandas
- Principales fonctionnalités de pandas pour la gestion d'opérations sur les séries et les data frames
  - Synthèse des principales fonctionnalités de pandas
- Quelques exemples illustratifs des fonctionnalités pandas appliquées sur des séries et des data frames
  - Création de séries et de data frames (listes, dictionnaires, fichiers)
  - Filtrage
  - Regroupement et Agrégation
- Mise en pratique de Pandas à travers des exercices simples sur Jupyter Notebooks

# Qu'est-ce que Pandas

- **Pandas** est une bibliothèque open-source en Python, largement utilisée pour la manipulation et l'analyse de données. Elle est construite sur **NumPy** et fournit des structures de données et des outils performants pour travailler avec des données tabulaires et structurées.
- Principales utilisations de pandas :
  - **Lire et écrire des données** dans différents formats (CSV, Excel, SQL, JSON, etc.).
  - **Manipuler des données** (filtrage, transformation, agrégation, etc.)
  - **Nettoyer des données** (traitement des valeurs manquantes, gestion des doublons, etc.).
  - **Effectuer des analyses statistiques descriptives** (moyennes, médianes, corrélations, etc.).

# Structures de données gérées par pandas (Series et Data Frames)

- Une **série** est une structure unidimensionnelle qui ressemble à un tableau (ou une liste). C'est l'équivalent d'une colonne de données dans une feuille de calcul ou une base de données.
- Caractéristiques principales des séries pandas:
  - **Indexées**: chaque élément d'une série est associé à une étiquette (un index) qui peut être utilisé pour accéder aux valeurs.
  - **Homogènes**: contient un seul type de données (entiers, chaînes de caractères, flottants, etc.).
  - **Equivalence à Numpy**: Une série pandas peut être vue comme un tableau Numpy avec des étiquettes d'index.

# Structures de données gérées par pandas (Series et Data Frames)

- Un **DataFrame** est une structure bidimensionnelle, similaire à une feuille de calcul Excel ou à une table dans une base de données. Il est composé de plusieurs séries (colonnes) ayant chacune son propre type de données.
- Caractéristiques principales des DataFrames pandas:
  - **Indexés**: les lignes et les colonnes sont étiquetées avec des indices.
  - **Agrégation de séries**: chaque colonne est une série, mais les colonnes peuvent avoir des types de données différents.
  - **Hétérogénéité**: les data frames pandas peuvent contenir des données hétérogènes (par exemple, des nombres, des chaînes de caractères, des booléens, etc.).

# Principales attributs des séries et des Data frames en pandas

## Principales attributs des séries Pandas

Attribut	Description
<code>index</code>	Renvoie les indices de la série sous forme d'un objet Index.
<code>values</code>	Renvoie les valeurs contenues dans la série sous forme de tableau NumPy.
<code>name</code>	Nom de la série (chaîne de caractères), peut être défini ou modifié.
<code>dtype</code>	Type de données des éléments de la série (int64, float64, object, etc.).
<code>size</code>	Nombre total d'éléments dans la série.
<code>shape</code>	Forme de la série (nombre de lignes, puisque c'est unidimensionnel).
<code>nbytes</code>	Quantité de mémoire utilisée par la série en octets.
<code>is_unique</code>	Indique si les valeurs de la série sont uniques (True ou False).
<code>is_monotonic</code>	Indique si les valeurs de la série sont monotones croissantes.
<code>empty</code>	Indique si la série est vide (True ou False).
<code>hasnans</code>	Indique la présence de valeurs manquantes (NaN) dans la série.

## Principales attributs des Data Frames Pandas

Attribut	Description
<code>index</code>	Renvoie les indices du DataFrame sous forme d'un objet Index.
<code>columns</code>	Renvoie les noms des colonnes sous forme d'un objet Index.
<code>values</code>	Renvoie les données du DataFrame sous forme de tableau NumPy.
<code>dtypes</code>	Renvoie les types de données de chaque colonne du DataFrame.
<code>shape</code>	Forme du DataFrame sous forme de tuple (nombre de lignes, nombre de colonnes).
<code>size</code>	Nombre total d'éléments dans le DataFrame (lignes * colonnes).
<code>empty</code>	Indique si le DataFrame est vide (True ou False).
<code>ndim</code>	Nombre de dimensions du DataFrame (toujours 2).
<code>axes</code>	Renvoie une liste des axes (l'index et les colonnes).
<code>T</code>	Transpose le DataFrame (inverse les lignes et les colonnes).

# Indexation et découpage en pandas

- **Indexing** (indexation) en Pandas fait référence à la manière d'accéder à des éléments spécifiques d'une série ou d'un DataFrame en utilisant des **étiquettes d'index** ou des **positions**. Cela permet d'accéder à des lignes, des colonnes ou des éléments individuels.
- **Slicing** (découpage) consiste à extraire une sous-partie des données, généralement un sous-ensemble de lignes ou de colonnes, en utilisant une plage d'étiquettes ou d'indices.



# Indexation et découpage en pandas

<i>Opération</i>	<i>Description</i>
<code>df['colonne']</code>	Sélectionne une colonne spécifique par son nom (retourne une Series).
<code>df[['colonne1', 'colonne2']]</code>	Sélectionne plusieurs colonnes en spécifiant une liste de noms de colonnes.
<code>df.iloc[0]</code>	Sélectionne la première ligne du DataFrame en utilisant l'index numérique.
<code>df.iloc[0:3]</code>	Sélectionne les lignes de la 0 à la 2 (slicing avec indices numériques).
<code>df.loc['index']</code>	Sélectionne une ligne spécifique en utilisant une étiquette d'index.
<code>df.loc['index1':'index3']</code>	Sélectionne plusieurs lignes par une plage d'étiquettes d'index (inclusif).
<code>df.iloc[:, 0]</code>	Sélectionne toutes les lignes de la première colonne (par indice numérique de colonne).
<code>df.loc[:, 'colonne']</code>	Sélectionne toutes les lignes d'une colonne par son nom (slicing par étiquette).
<code>df.iloc[0:3, 0:2]</code>	Sélectionne un sous-ensemble de lignes et colonnes en utilisant des indices numériques.
<code>df.loc['index1':'index3', 'colonne1':'colonne2']</code>	Sélectionne un sous-ensemble de lignes et de colonnes en utilisant des étiquettes (slicing étiquettes).
<code>df.at['index', 'colonne']</code>	Sélectionne un seul élément en utilisant une étiquette de ligne et une étiquette de colonne.
<code>df.iat[0, 1]</code>	Sélectionne un seul élément en utilisant des indices numériques pour la ligne et la colonne.
<code>df[df['colonne'] &gt; 5]</code>	Filtre les lignes où les valeurs de la colonne spécifiée sont supérieures à une condition.
<code>df.loc[df['colonne'] == 'valeur']</code>	Filtre les lignes où les valeurs de la colonne sont égales à une valeur spécifique.
<code>df.iloc[-1]</code>	Sélectionne la dernière ligne du DataFrame en utilisant l'index négatif.
<code>df['colonne'].iloc[0:5]</code>	Sélectionne les 5 premières lignes d'une colonne spécifique.
<code>df.iloc[:, -1]</code>	Sélectionne toutes les lignes de la dernière colonne.
<code>df.loc[df['colonne'].isin([1, 2])]</code>	Sélectionne les lignes où les valeurs de la colonne appartiennent à une liste donnée (condition multiple).

# Principales fonctionnalités de pandas pour la gestion d'opérations sur les séries et les data frames

- Pandas possède un grand nombre de fonctionnalités pour la gestion de diverses opérations qui peuvent être appliquées sur les séries et les data frames. Ces fonctionnalités peuvent être classifiées par thématiques:
  - Lecture/Écriture de fichiers : Importation et exportation de données.
  - Sélection et filtrage : Extraction de données spécifiques.
  - Manipulation et transformation de données : Transformation et fusion de DataFrames.
  - Gestion des valeurs manquantes : Traitement des valeurs nulles (NaN)
  - Opérations sur les colonnes : Gestion des colonnes et de l'index.
  - Statistiques et agrégation : Calculs statistiques et regroupement.
  - Fonctions avancées : Fonctions plus complexes pour la manipulation des données.

# Synthèse des principales fonctionnalités de pandas

## Lecture et écriture de fichiers

Fonction	Appel	Description
<code>read_csv()</code>	<code>pd.read_csv('fichier.csv')</code>	Lit un fichier CSV et renvoie un Data Frame.
<code>read_excel()</code>	<code>pd.read_excel('fichier.xlsx')</code>	Lit un fichier xlsx et renvoie un Data Frame
<code>to_csv()</code>	<code>df.to_csv('fichier.csv')</code>	Exporte le Data Frame vers un fichier CSV.
<code>to_excel()</code>	<code>df.to_excel('fichier.xlsx')</code>	Exporte le Data Frame vers un fichier Excel.

## Sélection et filtrage

Fonction	Appel	Description
<code>head()</code>	<code>df.head(n)</code>	Affiche les premières `n` lignes d'un Data Frame.
<code>tail()</code>	<code>df.tail(n)</code>	Affiche les dernières `n` lignes d'un Data Frame.
<code>iloc[]</code>	<code>df.iloc[i, j]</code>	Sélectionne des données par position (index numérique).
<code>loc[]</code>	<code>df.loc['index_label', 'col']</code>	Sélectionne des données par étiquette (nom de ligne/colonne).
<code>query()</code>	<code>df.query('condition')</code>	Filtre les lignes d'un Data Frame en utilisant une condition exprimée sous forme de chaîne de caractères (requête SQL-like).

## Opérations sur les colonnes et les index

Fonction	Appel	Description
<code>info</code>	<code>df.info()</code>	Affiche un résumé concis du Data Frame, incluant les types de données et les valeurs nulles
<code>columns</code>	<code>df.columns</code>	Renvoie les noms des colonnes du Data Frame.
<code>index</code>	<code>df.index</code>	Renvoie l'index (étiquettes des lignes) du Data Frame.
<code>astype()</code>	<code>df['colonne'].astype(dtype)</code>	Change le type de données d'une colonne.
<code>set_index()</code>	<code>df.set_index('colonne')</code>	Définit une colonne comme index du Data Frame.
<code>reset_index()</code>	<code>df.reset_index()</code>	Réinitialise l'index du Data Frame, en le ramenant à l'index par défaut (numérique).

## Gestion de données manquantes

Fonction	Appel	Description
<code>fillna()</code>	<code>df.fillna(valeur)</code>	Remplace les valeurs manquantes ('NaN') par une valeur donnée.
<code>isna()</code>	<code>df.isna()</code>	Renvoie un Data Frame booléen indiquant les valeurs manquantes ('NaN').
<code>dropna()</code>	<code>df.dropna()</code>	Supprime les lignes ou colonnes contenant des valeurs manquantes.

## Manipulation et transformation de données

Fonction	Appel	Description
<code>drop()</code>	<code>df.drop('colonne', axis=1)</code>	Supprime des colonnes ou des lignes d'un Data Frame.
<code>groupby()</code>	<code>df.groupby('colonne')</code>	Grouper les données par une ou plusieurs colonnes et appliquer des opérations comme `sum()`, `mean()`, etc.
<code>merge()</code>	<code>pd.merge(df1, df2, on='col')</code>	Fusionne deux Data Frames en fonction d'une ou plusieurs colonnes.
<code>concat()</code>	<code>pd.concat([df1, df2], axis=0)</code>	Concatène deux ou plusieurs DataFrames le long des lignes ou des colonnes.
<code>sort_values()</code>	<code>df.sort_values('colonne')</code>	Trie les lignes d'un Data Frame par les valeurs d'une colonne.
<code>replace()</code>	<code>df.replace(a, b)</code>	Remplace des valeurs spécifiques dans un DataFrame par d'autres.
<code>pivot()</code>	<code>df.pivot(index, columns, values)</code>	Restructure un Data Frame en fonction d'un ou plusieurs index, colonnes et valeurs.
<code>melt()</code>	<code>pd.melt(df, id_vars, value_vars)</code>	Transforme un DataFrame large en un format long.

## Statistiques et agrégation

Fonction	Appel	Description
<code>describe()</code>	<code>df.describe()</code>	Fournit des statistiques descriptives des colonnes numériques du Data Frame.
<code>mean()</code>	<code>df['colonne'].mean()</code>	Calcule la moyenne de la colonne
<code>median()</code>	<code>df['colonne'].median()</code>	
<code>pivot_table()</code>	<code>df.pivot_table(values, index)</code>	Crée un tableau croisé dynamique à partir des données du Data Frame.
<code>value_counts()</code>	<code>df['colonne'].value_counts()</code>	Compte les occurrences uniques des valeurs dans une colonne.
<code>corr()</code>	<code>df.corr()</code>	Calcule la matrice de corrélation des colonnes numériques.
<code>cumsum()</code>	<code>df['colonne'].cumsum()</code>	Calcule la somme cumulée des valeurs sur une colonne.
<code>cumprod()</code>	<code>df['colonne'].cumprod()</code>	Calcule le produit cumulé des valeurs sur une colonne.





# Synthèse des principales fonctionnalités de pandas

## Fonctions avancées

Fonction	Appel	Description
<code>apply()</code>	<code>df.apply(func)</code>	Applique une fonction à chaque élément d'une colonne ou ligne du DataFrame.
<code>duplicated()</code>	<code>df.duplicated()</code>	Renvoie un booléen indiquant si une ligne est dupliquée.
<code>drop_duplicates()</code>	<code>df.drop_duplicates()</code>	Supprime les lignes dupliquées dans un DataFrame.
<code>resample()</code>	<code>df.resample('freq')</code>	Regroupe les données selon une certaine fréquence (par exemple, pour des données temporelles).
<code>rolling()</code>	<code>df['colonne'].rolling(window=n)</code>	Applique des calculs sur des fenêtres glissantes de longueur définie (par exemple, moyenne mobile).
<code>expanding()</code>	<code>df['colonne'].expanding()</code>	Applique des calculs cumulatifs en considérant toutes les valeurs jusqu'à un point donné.
<code>applymap()</code>	<code>df.applymap(func)</code>	Applique une fonction élémentaire à chaque cellule d'un DataFrame.
<code>nunique()</code>	<code>df['colonne'].nunique()</code>	Renvoie le nombre de valeurs uniques dans une colonne.

# Quelques exemples illustratifs des fonctionnalités Pandas appliquées sur des séries et des data frames (Création de séries et de data frames)

## Tableau d'exemples introductifs à Pandas

Exemple	Lien html
Création de série à partir d'une liste aléatoire	
Création d'un data frame à partir d'un dictionnaire et changement d'indices	
Création d'un data frame à partir d'un fichier csv et changement d'indices	
Méthodologies de filtrage d'un data frame	
Regroupement et agrégation dans un data frame	