

TP 3 Pandas

Exercice 1 :

Datasets :

1. employees.csv :

Colonnes : employee_id, name, department, salary, hire_date, manager_id

2. projects.csv :

Colonnes : project_id, project_name, department, budget

3. work_hours.csv :

Colonnes : employee_id, project_id, hours

Questions :

1. Sélection avancée

Objectif : Trouver les employés qui ont été embauchés avant 2019 et qui ne travaillent pas dans le département "HR".

Indice : Utilisez des conditions multiples.

2. Groupement et agrégations

Objectif : Calculer le salaire moyen par département et trouver le département avec le salaire moyen le plus élevé.

Indice : Utilisez .groupby() et .agg().

3. Tris et filtrage avancé

Objectif : Trier les projets par budget décroissant, puis afficher les 2 projets avec les budgets les plus élevés.

Indice : Utilisez .sort_values() et .head().

4. Jointures simples

Objectif : Lister les noms des employés et les noms des projets sur lesquels ils ont travaillé, en joignant les tables employees et work_hours avec projects.

Indice : Utilisez .merge() avec plusieurs tables.

5. Agrégation sur plusieurs tables

Objectif : Calculer le nombre total d'heures travaillées par département, puis classer les départements par ordre décroissant d'heures totales.

Indice : Combinez les tables employees, work_hours, et projects pour obtenir le résultat.

6. Création de colonnes personnalisées

Objectif : Ajouter une colonne bonus dans la table employees, où le bonus est calculé comme 10% du salaire pour les employés embauchés avant 2018.

Indice : Utilisez .apply() ou .loc() pour calculer cette colonne.

Exercice 2 : Filtrage et Sélection

Dataset : students.csv

Questions :

1. **Filtrer les étudiants** avec une note supérieure à 80.
2. Trouver **les étudiants de Computer Science** diplômant en 2024.
3. Ajouter une colonne calculant une **mention** basée sur la note (grade), avec :
 - "Excellent" si $\text{grade} \geq 90$
 - "Bien" si $80 \leq \text{grade} < 90$
 - "Passable" sinon.
4. Trier les étudiants par grade en ordre décroissant.
5. Calculer la **moyenne des notes par spécialité (major)**.

Exercice 3 : Jointures et Agrégations

Datasets :

1. **Dataset** : orders.csv

Colonnes : order_id, customer_id, product_id, quantity, order_date

2. **Dataset** : products.csv

Colonnes : product_id, product_name, category, price

Questions :

1. Joindre les deux tables pour obtenir les commandes avec le product_name et le category.
2. Calculer les **ventes totales (quantity * price) par produit**.
3. Trouver le **nombre de commandes par client** (customer_id).

4. Afficher les **produits qui n'ont jamais été commandés** (jointure droite ou anti-jointure).
5. Calculer le **total des ventes par catégorie**.

Exercice 4 :

Datasets : employees_tp3.csv, departments.csv et salaries.csv

Questions :

1. Trouver les employés et leurs départements (department_name).
2. Calculer le **salaire total (salaire + bonus)** par employé.
3. Lister les employés travaillant dans un lieu (location) particulier.
4. Identifier les employés dont le salaire est supérieur à la moyenne de leur département.
5. Afficher les départements qui n'ont pas d'employés associés.