



Data Management
DATA MANAGEMENT



ENSIAS
LECTURE ACADÉMIQUE

ENSIAS · UNIVERSITÉ MOHAMMED V DE RABAT · 1ÈRE ANNÉE · GÉNIE DE LA DATA

Initiation au management de la data

TP 2

Pipeline et indicateurs

De la collecte à la décision : rendre les données fiables, gouvernées et utiles



Data Management
DATA MANAGEMENT

TP · SOMMAIRE

Table des matières

Mission	3
Télécharger et ouvrir le kit	3
Lignes de code à compléter dans le notebook	4
Déroulé des 2 heures	6
Objectifs et contrat de pipeline	6
Questions à traiter dans le rapport	6
Contrat des fichiers produits	7
Livrables à remettre	7
Remise et critères de réussite	7





TP2 · data-management · Version publiée

Durée : 2 heures — travail individuel. Chaque étudiant construit et documente un pipeline court à partir du jeu de données fourni dans ce kit.

Ce TP transforme des données nettoyées en indicateurs utiles à une décision, puis explicite ce que ces indicateurs permettent — ou ne permettent pas — de conclure.

Mission

Vous êtes chargé de préparer une note courte pour le responsable énergie du campus. Un fichier nettoyé, son dictionnaire, son journal et une matrice de gouvernance sont fournis dans l'archive. Votre mission est de construire une chaîne reproductible — chargement, contrôle, transformation, indicateurs, visualisation et décision — et de répondre à la question : **quels bâtiments faut-il examiner en priorité, compte tenu du volume observé et de la couverture des mesures ?**

Entrées fournies dans l'archive : `data/donnees_nettoyees_reference.csv`,
`data/dictionnaire_donnees_reference.csv`,
`data/controle_qualite_reference.json`,
`data/journal_transformations_reference.csv` et
`data/matrice_gouvernance_reference.md`.

Le fichier `matrice_gouvernance_modele.md` est à compléter et à remettre sous le nom `matrice_gouvernance_completee.md`.

Sorties obligatoires : pipeline exécutable, deux indicateurs documentés, deux visualisations enregistrées et recommandation argumentée.

Télécharger et ouvrir le kit

Téléchargez depuis la page du TP l'énoncé PDF et le **Code de départ (.zip)**. Extrayez le ZIP dans un dossier personnel ; le kit contient le notebook, le modèle de compte rendu et les données de référence.

Linux/macOS :

TERMINAL · BASH

```
mkdir -p "$HOME/ensias-tps/Data-Management/TP2_NOM_PRENOM"
unzip "<nom-exact-du-kit-TP2>.zip" -d "$HOME/ensias-tps/Data-Management/TP2_NOM_PRENOM"
cd "$HOME/ensias-tps/Data-Management/TP2_NOM_PRENOM"
```

Windows PowerShell :

CODE

```
$work = "$HOME\ensias-tps\Data-Management\TP2_NOM_PRENOM"  
New-Item -ItemType Directory -Force $work  
Expand-Archive -LiteralPath ".\<nom-exact-du-kit-TP2>.zip" -DestinationPath $work  
Set-Location $work
```

Depuis le dossier extrait, créez l'environnement puis lancez JupyterLab :

TERMINAL · BASH

```
python3 -m venv .venv  
source .venv/bin/activate  
python -m pip install --upgrade pip  
python -m pip install -r requirements.txt  
python -m jupyter lab
```

Sous Windows PowerShell, activez avec `.\.venv\Scripts\Activate.ps1` avant les commandes `python -m pip`. Ouvrez `tp02_pipeline_indicateurs.ipynb` et exécutez les cellules dans l'ordre. Le notebook utilise les fichiers locaux du dossier `data/` et affiche la source retenue.

Lignes de code à compléter dans le notebook

Chaque partie pratique du notebook contient au moins une vraie tâche de programmation. Les **TODO CODE** demandent de compléter une fonction courte ou une ou plusieurs instructions Python. Les **TODO PARAMETRE** proposent en plus des valeurs simples à modifier ; les valeurs de départ restent valides.

The logo of ENSIAS, featuring the word "ENSIAS" in a stylized, light red font, with a large, faint red triangle in the background.

Partie	Ligne à compléter	Résultat attendu
Partie 1 — préparation	fonction <code>lister_fichiers</code>	retourner les fichiers du dossier
Partie 2 — question	fonction <code>nettoyer_libelle</code>	normaliser un libellé texte
Partie 3 — total	fonction <code>filtrer_mesures_valides</code>	conserver les lignes utilisables
Partie 4 — moyenne	fonction <code>calculer_moyenne</code>	calculer une moyenne par groupe
Partie 5 — contrôles	fonction <code>controle_bloquant</code>	identifier un contrôle bloquant
Partie 6 — preuves	fonction <code>lire_preuve_texte</code>	lire une preuve documentaire
Partie 7 — classements	fonction <code>ligne_ecart_maximal</code>	repérer l'écart de rang maximal
Partie 8 — visualisations	fonction <code>enregistrer_figure</code>	enregistrer une figure
Partie 9 — exports	fonction <code>fichiers_absents</code>	repérer les fichiers manquants
Partie 10 — restitution	fonction <code>formuler_recommandation</code>	produire une phrase de recommandation

Pour chaque fonction, remplacez `pass` par une ou plusieurs instructions Python, exécutez la cellule puis relancez la cellule finale de vérification. Les fonctions doivent rester courtes et utiliser les variables déjà préparées dans le notebook. Par exemple, `calculer_moyenne` doit utiliser `groupby` et `formuler_recommandation` doit retourner une phrase qui cite le type de recommandation et l'indicateur choisi.

Les agrégations, graphiques et contrôles principaux sont fournis pour guider le travail. Les fonctions marquées `TODO CODE` sont à compléter directement dans le notebook, avec l'indice placé sous chaque fonction.

Chaque partie contient une cellule `TODO CODE` proposant une complétion de fonction ou de lignes Python. Les paramètres `TODO PARAMETRE` restent simples et les valeurs initiales sont déjà valides. La cellule finale appelle les fonctions et vérifie les fichiers produits ; une dernière cellule permet d'activer le contrôle strict avant remise.

Déroulé des 2 heures

Temps	Travail	Trace attendue
10 min	reprise des entrées et du périmètre	source et nombre de lignes
20 min	contrat du pipeline	étapes, contrôles et règles du dossier fourni
35 min	calcul des indicateurs	formules, unités, périodes et dénominateurs
20 min	visualisations et contrôles	deux figures enregistrées et vérifiées
20 min	interprétation	limites, gouvernance et recommandation
15 min	finalisation	rapport et archive de remise

Objectifs et contrat de pipeline

- assembler chargement, contrôle, transformation, indicateurs et export dans un pipeline lisible ;
- calculer au moins deux indicateurs cohérents avec une question de décision ;
- produire une visualisation lisible par indicateur ;
- relier les résultats au dictionnaire, au rapport qualité et à la gouvernance ;
- expliciter les limites, les risques d'interprétation et une recommandation.

Le pipeline doit laisser une preuve à chaque frontière : fichier d'entrée, nombre de lignes chargées, contrôles bloquants, formule de chaque indicateur, période, dénominateur, fichiers exportés et recommandation. Une figure affichée à l'écran mais non enregistrée ne constitue pas un livrable.

Questions à traiter dans le rapport

1. Quels fichiers sont disponibles et quelle source le notebook a-t-il réellement utilisée ?
2. Pourquoi faut-il préciser l'unité, la période et le nombre de mesures ?
3. Quel bâtiment a le plus grand total et que ne permet pas ce résultat de conclure ?
4. Le classement par moyenne est-il identique au classement par total ? Quel est le rôle du nombre de mesures ?
5. Quel contrôle est bloquant et quelle limite reste présente après les contrôles ?
6. Quelle transformation et quelle décision qualité sont documentées ?
7. Pour quel bâtiment l'écart entre les rangs est-il le plus grand ? Quelle information supplémentaire demander ?
8. Que montrent les deux graphiques et que ne permettent-ils pas de conclure ?
9. Quelle action recommandez-vous, quelle vérification doit la précéder et quelle preuve devra être conservée ?

Contrat des fichiers produits

Les colonnes minimales attendues sont :

- `donnees_finales.csv` : les lignes effectivement utilisées, avec les colonnes de mesure et la provenance de la source ;
- `indicateurs.csv` : un enregistrement par bâtiment avec `consommation_totale_kwh`, `consommation_moyenne_kwh` et `nombre_mesures` ;
- `fiche_indicateurs.csv` : nom, formule, unité, période, périmètre, dénominateur et limite de chaque indicateur ;
- `comparaison_rangs.csv` : classement par volume total, moyenne par mesure et écart de classement ;
- `pipeline_manifest.json` : entrée, lignes utilisées, transformations et sorties produites ;
- `matrice_gouvernance_completee.md` : responsabilités, accès, risque, protection, conservation et vérification préalable ;
- `visualisations/` : deux images nommées et lisibles, une par indicateur.

Une colonne calculée sans définition dans la fiche d'indicateurs est considérée comme incomplète.

Livrables à remettre

1. `tp02_pipeline_indicateurs.ipynb` exécuté avec sortie visible ;
2. `rapport_tp02.md` complété à partir de `rapport_tp02_modele.md` ;
3. `donnees_finales.csv` et `indicateurs.csv` ;
4. `fiche_indicateurs.csv` et `comparaison_rangs.csv` ;
5. deux fichiers image dans `visualisations/` ;
6. `pipeline_manifest.json` et `matrice_gouvernance_completee.md` ;
7. le dictionnaire, le contrôle qualité, le journal et la matrice de gouvernance de référence fournis dans `data/` ;
8. une recommandation d'une demi-page.

Le rapport doit répondre à la question d'analyse choisie, donner les résultats avec leurs unités et périodes, interpréter les deux figures et signaler les limites de couverture, de qualité et de comparabilité.

Le notebook corrigé enseignant est disponible dans `teacher/tp02_pipeline_indicateurs_corrige.ipynb`.

Remise et critères de réussite

Regrouper les fichiers dans une archive nommée `nom_prenom_filiere.zip` (sans espace ni accent). Utiliser comme objet : `Data Management – TP2 – Nom Prénom – Filière`.

Critère	Attendu
Pipeline	les étapes s'exécutent dans l'ordre et la source est affichée
Indicateurs	formule, unité, période, périmètre et dénominateur sont explicites
Visualisations	deux figures sont enregistrées, titrées et sourcées
Traçabilité	les fichiers d'entrée et les exports sont identifiables
Décision	la recommandation reste proportionnée aux preuves et aux limites

