

Cours d'analyse de données en géographie

Niveau Master 1 - GEANDO

Séance Bonus pour les niveaux débutants et intermédiaires

Maxime Forriez^{1,a}

¹ Institut de géographie, 191, rue Saint-Jacques, Bureau 105, 75 005 Paris,
^amaxime.forriez@sorbonne-universite.fr

7 novembre 2025

Courage, c'est presque fini !

1 Objectifs

- Gérer un A.P.I.
- Télécharger automatiquement des données mises à disposition sur un A.P.I.

2 Manipulations

L'exercice compte en bonus de la note finale pour les parcours « débutants » et « intermédiaires ».

L'exercice est relativement simple. Il s'agit juste de télécharger automatiquement des données à partir d'un *Application Programming Interface* (A.P.I.).

Les A.P.I. sont très nombreuses. Il suffit de consulter <https://www.data-gouv.fr/dataservices/search> pour s'en convaincre.

Ici, vous allez utiliser l'A.P.I. de <https://www.infoclimat.fr>.

1. Aller à l'adresse : <https://www.infoclimat.fr/opendata>

- Créer votre compte.
- À partir de l'adresse précédente, générer une clé (ou un jeton, ou un *token*).

N.B. N'oubliez pas de préciser la **non** utilisation commerciale.

- Zoomer sur Paris. Dans le cinquième arrondissement, cliquer sur la station [MAE]
LATMOS-PARIS
- Générer un lien de téléchargement comprenant votre clé.
- Essayer de comprendre la logique des mots clés de l'adresse U.R.L. générée, ici vous avez :
 - start=
 - end=
 - format=
 - stations []=
 - method=
 - token=

N.B. 1. Pour plus d'information sur les A.P.I. Rest : <https://www.data-bird.co/blog/api-rest>.

N.B. 1. Toutes les A.P.I. n'ont pas obligatoirement de *token*.

2. Pour télécharger automatiquement les données, il faut utiliser la bibliothèque `requests` et se référer à la documentation : https://rtavenar.github.io/poly_python/content/api.html
 - Avec la fonction fournie `geturl()` et avec la liste des variables demandées, créer une variable `url`. Vous devez télécharger les données entre le 1^{er} octobre et le 30 octobre de cette année.
 - Avec la méthode `requests.get(...)` et la fonction `testConnexion()`, télécharger les données de l'A.P.I. dans une variable `data`.
- Si la réponse est égale à 200, alors la fonction utilise l'attribut `.text` pour lire et télécharger les données en format C.S.V.
- Les données récupérées sont textuelles et mélangées. Il faut isoler les données, les titres des colonnes et les métadonnées en créant trois listes `data2`, `titre` et `metadata`. Créer une boucle avec des conditions permettant isoler les lignes 0, 1, 2, 3, 4, 6 pour les métadonnées.; la ligne 5 pour les titres, et le reste du tableau pour les données.

3. Convertir les listes `data2` et `metadata` en objet Pandas

- Créer un objet Pandas, en réaffectant la variable `data`, à partir de `data2`. Ajouter à cet objet le titre des colonnes en utilisant l'attribut `columns`
- Créer un objet Pandas à partir de `metadata`.
- Enregistrer l'objet `data` avec la méthode `to_excel()`
- Enregistrer l'objet `metadata` avec la méthode `to_csv()`. Il faudra préciser le paramètre `encoding="utf-8"`

Si tout s'est bien passé vous avez récupéré 28 champs :

1. `station_id`;

```
2. dh_utc;
3. temperature;
4. pression;
5. pression_variation_3h;
6. humidite;
7. point_de_rosee;
8. visibilite;
9. vent_moyen;
10. vent_rafales;
11. vent_rafales_10min;
12. vent_direction;
13. temperature_min;
14. temperature_max;
15. pluie_1h;
16. pluie_3h;
17. pluie_6h;
18. pluie_12h;
19. pluie_24h;
20. pluie_cumul_0h;
21. pluie_intensite;
22. pluie_intensite_max_1h;
23. uv;
24. uv_index;
25. ensoleillement;
26. temperature_sol;
27. temps_omm;
28. complements;source.
```

3 Bonus

Refaire l'exercice en téléchargeant les données à partir du format J.S.O.N.