

DS n° 02 – Production Éolienne/Solaire

- Faire tous les exercices dans un fichier NomPrenom.py à sauvegarder,
- mettre en commentaire l'exercice et la question traités (ex : # Exercice 1),
- ne pas oublier pas de commenter ce qui est fait dans votre code (ex : # Je crée une fonction pour calculer la racine d'un nombre),
- il est possible de demander un déblocage pour une question, mais celle-ci sera notée 0,
- il faut vérifier avant de partir que le code peut s'exécuter et qu'il affiche les résultats que vous attendez. Les lignes de code qui doivent s'exécuter sont décommentées.

1 Introduction

Depuis sa création, le 24 mars 2000, la Commission de régulation de l'énergie (CRE) veille au bon fonctionnement des marchés de l'électricité et du gaz en France, au bénéfice des consommateurs finaux et en cohérence avec les objectifs de la politique énergétique.



Une de ses attributions est de calculer le complément de rémunération. C'est un dispositif, introduit par la LTECV (Loi de Transition Énergétique pour la Croissance Verte), de soutien des producteurs d'énergie éolienne et solaire prévu aux articles L. 314-18 à L. 314-27 du code de l'énergie.

Dans ce mécanisme où les producteurs d'électricité à partir d'énergie renouvelable commercialisent leur énergie directement sur les marchés, une prime vient compenser l'écart entre les revenus tirés de cette vente et un niveau de rémunération de référence, fixé selon le type d'installations par la puissance publique dans le cadre d'un arrêté tarifaire ou par le producteur dans le cadre d'une procédure de mise en concurrence.



Ce complément de rémunération peut généralement être qualifié de prime variable, ou ex post, dans la mesure où son montant s'ajuste pour compenser la différence entre la rémunération de référence et un revenu marché de référence.

Ce dispositif vise à exposer les producteurs aux signaux des prix de marché de court terme, tout en leur garantissant une rémunération raisonnable.

Afin de calculer les prix de référence utilisés dans le calcul du complément de rémunération pour les filières éoliennes et solaires, le jeu de données suivant peut être utilisé. Il est constitué des chroniques de production mensuelle éolienne et solaire au pas horaire, à la maille France depuis 2020.

Il est disponible en suivant le lien suivant :

<https://www.data.gouv.fr/fr/datasets/courbes-de-production-mensuelles-eolien-solaire-complement-de-remuneration/>

Pour les besoin de cet exercice, les données sont disponibles sous la forme du fichier `production-solaire-eolienne.csv` dont les colonnes sont séparées par des points-virgules. Il est disponible dans le dossier « `/home/eleve/Ressources/PTSI/` ». Afin d'éviter des modifications intempestives du fichier, celui-ci ne devra pas être copié dans le dossier de travail mais consulté directement depuis le dossier d'origine.

2 Étude quotidienne

- Question 1** **Écrire** un code permettant de générer une *liste* lignes contenant l'ensemble des lignes du fichier `production-solaire-eolienne.csv`. **Afficher** les deux premiers éléments de la liste `lignes`.
- Question 2** **Écrire** un code permettant d'afficher une *liste* contenant les données de la ligne d'index 1, découpées selon le séparateur de colonnes. **Afficher** cette liste.
- Question 3** **Écrire** un code permettant de générer une liste `journee` contenant l'ensemble des *listes*, contenant les données des lignes découpées selon le séparateur de colonnes, correspondantes à la journée du *01 décembre 2022*. **Afficher** la liste `journee`.

Les trois premiers éléments de la liste `journee` sont :

- 1 `['2022-12-01', '07:00', '4270', '1']`,
- 2 `['2022-12-01', '09:00', '4071', '694']`,
- 3 `['2022-12-01', '10:00', '3751', '1483']`.

On constate que les informations **n'apparaissent pas dans l'ordre chronologique**, ni pour la date ni pour l'heure.

- Question 4** **Créer** une fonction `get_journee(date)` qui à partir d'une journée `date` (codée sous la forme d'une chaîne de caractères) retourne la liste `journee` correspondante à la journée `date`.

La commande `print(get_journee('2022-12-01'))` affiche le résultat de la question 3.

- Question 5** **Générer** deux listes `jour_eolien` et `jour_solaire` qui contiennent respectivement les valeurs de la production éolienne et solaire pour la journée du *01 décembre 2022* triées par ordre croissant d'heure (00,01,02,03,...,23).

Les trois premiers éléments de :

— jour_eolien sont : 3153, 3219, 3187

— jour_solaire sont : 2, 1, 1

Cela signifie que la production éolienne en France a été de 3153MWh le *01 décembre 2022* entre 0h et 1h, puis de 3219MWh entre 1h et 2h, tandis que la production solaire a été de 2MWh entre 0h et 1h, puis de 1MWh entre 1h et 2h.

Le code suivant permet de tracer la figure 1 :

```
import matplotlib.pyplot as plt

liste1=[0,0,0.1,.4,.5,.7,.8,.5,.5,.4,.5,.2\
        ,.3,0,.1,.1,0,.1,0,0,0,0,0]
liste2=[0,0,3,4,5,7,8,10,10.5,11,11.5,12,13\
        ,10,11,9,8,5,4,3,0,0,0]

plt.plot(liste1,label="Lac d'Ayous")
plt.plot(liste2,label="Pic du Midi d'Ossau")
plt.xticks(range(len(liste1)))
plt.legend()
plt.show()
```

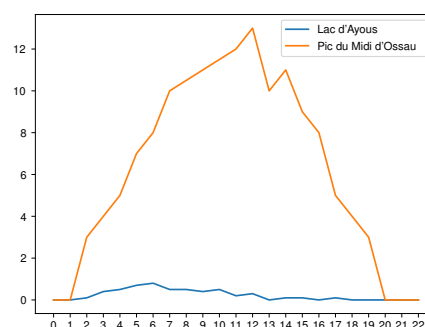


FIGURE 1 – Tracé Matplotlib

Question 6 A partir de cet exemple, **tracer** pour la journée du *01 décembre 2022* les courbes de la production éolienne et solaire.

La courbe du *01 novembre 2022* est donnée à titre d'exemple sur la figure 2.

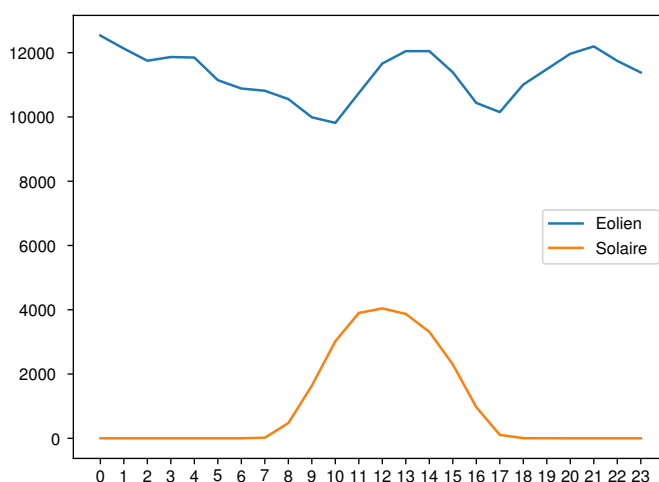


FIGURE 2 – Courbe pour le *01 novembre 2022*

Question 7 Tracer une nouvelle figure qui superpose les tracés des journées du *15 août 2022* et du *01 décembre 2022*. Il est imposé pour cela utiliser une boucle `for`. Le départ du script est donné dans la suite.

Départ du script à utiliser pour la question 7 :

```
jours_eolien=[[ ],[ ]]
jours_solaire=[[ ],[ ]]
dates=['2022-08-15','2022-01-01']
for id_jour in range(2):
    journee=get_journee(dates[id_jour])
```

La courbe pour les *01 novembre 2022* et *15 juillet 2022* est donnée à titre d'exemple sur la figure 3.

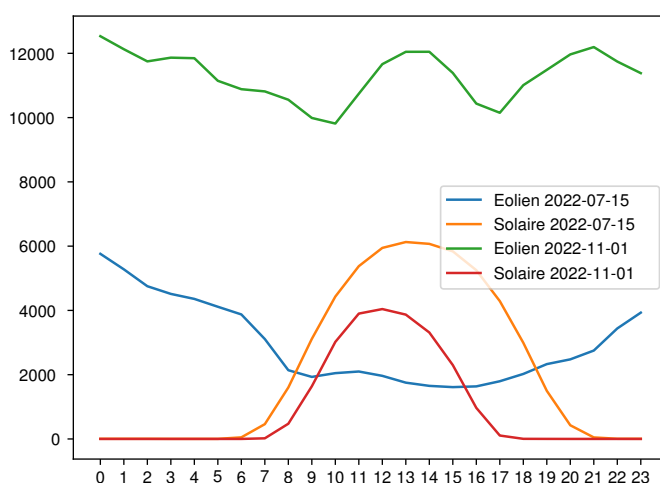


FIGURE 3 – Courbe pour les *01 novembre 2022* et *15 juillet 2022*

3 Étude mensuelle

On cherche maintenant à déterminer les productions mensuelles pour l'année 2022.

Question 8 Déterminer les listes `mois_eolien` et `mois_solaire` qui contiennent les cumuls des consommations mensuelles (somme de chaque jour du mois). Les listes devront être classées par mois croissant. Ainsi, `mois_eolien[0]` contiendra la production éolienne en France pour le mois de janvier (3140824MWh dans notre cas).

Le code suivant permet de tracer la figure 4 :

```
import matplotlib.pyplot as plt

liste1=[0,0,0.1,.4,.5,.7,.8,.5,.5,.4,.5,.2\
        ,.3,0,.1,.1,0,.1,0,0,0,0,0]
liste2=[0,0,3,4,5,7,8,10,10.5,11,11.5,12,13\
        ,10,11,9,8,5,4,3,0,0,0,]

labels=[i+1 for i in range(len(liste1))]
plt.bar(labels,liste2, label="Ossau")
plt.bar(labels,liste1, label="Lac d'Ayous")

plt.xticks(range(len(liste1)),labels=labels)
plt.legend()
plt.show()
```

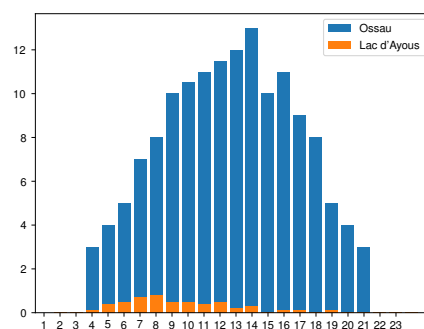


FIGURE 4 – Tracé Matplotlib

Question 9 A partir de cet exemple, **tracer** pour l'année 2022 les barres mensuelles de la production éolienne et solaire.

FIN

Correction

Question 1

```
file=open('production-solaire-eolienne.csv','r')
contenu=file.read()
file.close()
lignes=contenu.split('\n')
print(lignes[:2])
```

Question 2

```
print(lignes[1].split(';'))
```

Question 3

```
journee=[]
for ligne in lignes:
    data=ligne.split(';')
    if data[0]=='2022-12-01':
        journee.append(data)
```

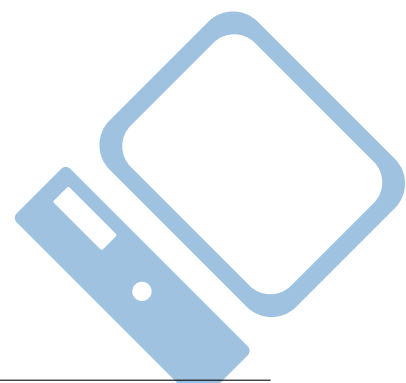
Question 4

```
def get_journee(date):
    journee=[]
    for ligne in lignes:
        data=ligne.split(';')
        if data[0]==date:
            journee.append(data)
    return journee
```

Question 5

```
jour_eolien=[]
jour_solaire=[]
journee=get_journee('2022-08-15')
for i in range(24):
    for heure in journee:
        if i==int(heure[1][:2]):
            jour_eolien.append(int(heure[2]))
            jour_solaire.append(int(heure[3]))
```

Question 6



Correction

```
import matplotlib.pyplot as plt

plt.plot(jour_eolien, label='Eolien')
plt.plot(jour_solaire, label='Solaire')
plt.xticks(range(24))
plt.legend()
plt.show()
```

Question 7

```
jours_eolien=[[],[]]
jours_solaire=[[],[]]
dates=['2022-08-15','2022-01-01']
for id_jour in range(2):
    journee=get_journee(dates[id_jour])
    for id_heure in range(24):
        for heure in journee:
            if id_heure==int(heure[1][:2]):
                jours_eolien[id_jour].append(int(heure[2]))
                jours_solaire[id_jour].append(int(heure[3]))

plt.plot(jours_eolien[id_jour], label='Eolien '+dates[id_jour])
plt.plot(jours_solaire[id_jour], label='Solaire '+dates[id_jour])

plt.xticks(range(24))
plt.legend()
plt.show()
```

Question 8

```
mois_eolien=[0]*12
mois_solaire=[0]*12
for ligne in lignes:
    data=ligne.split(';')
    if data[0][:4]=='2022':
        if data[2]:
            mois_eolien[int(data[0][5:7])-1]+=int(data[2])
        if data[3]:
            mois_solaire[int(data[0][5:7])-1]+=int(data[3])
```

Question 9

```
plt.bar([i+1 for i in range(12)],mois_eolien, label='Eolien')
plt.bar([i+1 for i in range(12)],mois_solaire, label='Solaire')
plt.xticks(range(12),labels=[i+1 for i in range(12)])
plt.legend()
plt.show()
```