

PROJET PYTHON

Sujet : Les émissions de gaz à effet de serre et l'écotaxe

De : James Kennedy

Nous avons à disposition les bases de données :

- Gas_effet_serre : valeurs trimestrielles 2015Q1 - 2020Q4
- Population : valeurs annuelles 2015 - 2020
- PIB : valeurs annuelles 2015 - 2020
- Recettes fiscales environnementales : valeurs annuelles 2015 - 2020

IMPORTATION

Pour plus de facilité et pour rester dans le cadre du cours, nous avons décidé de convertir nos fichiers *xlsx* en fichiers *txt* et de les réorganiser.

1. Méthode d'utilisation et d'organisation d'un fichier

Exemple sur le fichier *Gas_effet_serre2015-2020*.

- On ouvre et on lit le fichier.

```
file_gas = open("chemin\\Gas_effet_serre2015-2020.txt", "r") # sur Spyder
```

```
file_gas = open("Gas_effet_serre2015-2020", "r") # sur Pycharm
```

- On met toutes les lignes dans une liste.

```
lines_gas = file_gas.readlines()
```

Chaque élément de la liste représente une ligne du fichier texte.

Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

- Maintenant, que l'on sait à quoi ressemblent nos données, on peut les réorganiser.

```
number_lines = len(lines_gas)
file_gas = open("Gas_effet_serre2015-2020", "r")
# comme on ne peut lire le fichier qu'une fois, il faut relancer la commande
Gas_effet_serre = []
# cette liste vide contiendra les données finales lorsqu'elles seront séparées
i = 0 # point de départ des itérations

while i < number_lines :

    line = file_gas.readline() # stockage de la première ligne dans une liste
    line = line[:-1] # on supprime le caractère de retour à la ligne ("\n")
    string = line.split(" ")
    # on sépare les éléments lorsqu'il y a un espace dans la chaîne de caractère

    x = 1
    while x < len(string) :
        string[x] = float(string[x])
        # on transforme tous les éléments de la liste (qui sont en chaîne de caractère)
        # en float (nombre à virgule) sauf l'élément 0 qui est le nom du pays
        # si la base contient des nombres sans virgule
        # on utilise int() (nombre)
        x = x + 1

    Gas_effet_serre.append(string) # on ajoute la liste obtenue dans la liste data
    i = i + 1 # on passe à la ligne suivante
```

On obtient une grande liste contenant des sous-listes.

Chaque pays et ses données constituent une sous-liste.

commande utile : *nom_liste[index_sous_liste][index_element]*

PARTIE 1

1. Regroupement des données trimestrielles en données annuelles

Avec la méthode décrite ci-dessus, on récupère les données sur les émissions de gaz à effet de serre (fichier *Gas_effet_serre2015-2020*).

- On définit au préalable les variables utiles puis à l'aide de deux boucles *while* on extrait les données pour les ajouter dans *Gas_effet_serre_annuel* après les avoir sommées.

```
Gas_effet_serre_annuel = []
pays = 0
a = 1
somme = 0
while pays <= len(Gas_effet_serre) - 1 : # 27 pays - 1
    Gas_effet_serre_annuel.append(Gas_effet_serre[pays][0])
    # on ajoute la valeur à l'index 0 de chaque sous liste (le nom du pays)

    while a <= len(Gas_effet_serre[0]) - 4 :
        somme = somme + Gas_effet_serre[pays][a] + Gas_effet_serre[pays][a+1] +
            Gas_effet_serre[pays][a+2] + Gas_effet_serre[pays][a+3]
        Gas_effet_serre_annuel.append(somme)
        # on ajoute dans la liste la somme des 4 trimestres
        somme = 0
        a = a + 4 # on recommence pour l'année suivante

    pays = pays + 1
    a = 1 # pour qu'à chaque itération pays += 1, on commence avec a = 1
```

Gas_effet_serre_annuel est une très grande liste.

Elle ne comporte pas de sous-liste comme pour *Gas_effet_serre*.

Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

2. Tri des pays en fonction de leurs émissions de gaz à effet de serre en 2020

- Préparation.

```
i = 0 # on démarre l'itération
j = 6 # pour avoir la valeur de 2020, on prend la valeur qui a la 6e position après
le nom du pays
pays = []
annee_2020 = []

while i <= len(Gas_effet_serre_annuel) - 7 :
    # - 7 car il y a 7 valeurs pour chaque pays (avec le nom compris)
    pays.append(Gas_effet_serre_annuel[i]) # on ajoute le nom du pays
    annee_2020.append(Gas_effet_serre_annuel[j]) # on ajoute la valeur pour
l'année 2020
    i = i + 7
    j = j + 7 # on passe au pays suivant
```

On obtient deux listes, l'une contenant les noms des pays et l'autre les valeurs de 2020 pour chacun d'eux.

- On crée un fichier qui contiendra les valeurs triées.

```
classement_émission_gaz = open('Part1_Question2.txt', 'w')
# 'w' : crée le fichier puisqu'il n'existe pas encore
classement_émission_gaz = open('Part1_Question2.txt', 'r+')
# 'r+' : permet de lire et d'écrire dans le fichier
```

- On fait le tri.

```
maximum = []
liste_max = [] # cette liste nous servira pour la question bonus

while annee_2020 != [] :
    i = annee_2020.index(max(annee_2020))
    # i prend la valeur de l'index où se situe la valeur maximum
    maximum.append(pays[i] + " : " + str(max(annee_2020)) + "\n")
    # cette liste contient une phrase pour chaque pays et son émission
```

Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

```
liste_max.append(pays[i])
liste_max.append(max(annee_2020))
# liste contient le nom du pays, suivie de son émission en 2020

annee_2020.pop(i)
pays.pop(i)
# on supprime le maximum (l'année et le pays)
# pour avoir un autre maximum à la prochaine itération
```

- On ajoute le résultat dans le fichier texte.

```
classement_émission_gaz.writelines(maximum) # sur Spyder
classement_émission_gaz.write(maximum) # sur Pycharm
```

3. Quel pays de l'UE a produit le moins et le plus d'émissions de gaz à effet de serre en 2020 ?

Au préalable, on refait la même préparation qu'à la question 2 pour obtenir les listes *pays* et *annee_2020*.

- Maximum

```
def max_émission_gaz():
    maximum = max(annee_2020) # on utilise la fonction built-in max()
    i = annee_2020.index(maximum)
    print("Le pays de l'UE ayant produit le plus d'émissions de gaz ",
          "à effet de serre en 2020 est : ", pays[i], ", avec un total de ",
          maximum, " mille tonnes. ")
```

- Minimum

On fait de même, mais avec la fonction built-in *min()*.

4. Calcul de la moyenne pondérée des émissions de gaz à effet de serre de l'UE en 2015 et 2020.

On répète la même [méthode](#) que dans la question 1, mais avec les données sur la population (fichier *Population2015-2020*).

Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

- On définit des variables qui vont nous servir pour nos boucles *while*.

```
i = 0
annee_2015 = 1 # la première donnée relative à l'année 2015 se trouve
# à la position 1 (2e valeur) de notre liste Gas_effet_serre_annuel
annee_2020 = 6 # car la première donnée relative à l'année 2020 se trouve
# à la position 6 (7e valeur) de notre liste Gas_effet_serre_annuel
gaz_2015 = 0
nb_habitants_2015 = 0
gaz_2020 = 0
nb_habitants_2020 = 0
```

- On récupère le nombre d'habitants.

```
while i <= len(Population) - 1 :
    nb_habitants_2015 = nb_habitants_2015 + Population[i][annee_2015]
    nb_habitants_2020 = nb_habitants_2020 + Population[i][annee_2020]
    i = i + 1
```

- On récupère les gaz à effet de serre des deux années.

```
while annee_2015 <= len(Gas_effet_serre_annuel) - 6 :
    # on soustrait 6 car cela nous donne la position de notre dernière donnée relative
    # à l'année 2015
    gaz_2015 = gaz_2015 + Gas_effet_serre_annuel[annee_2015]
    gaz_2020 = gaz_2020 + Gas_effet_serre_annuel[annee_2020]
    annee_2015 = annee_2015 + 7
    annee_2020 = annee_2020 + 7
```

- On calcul les moyennes et on les arrondit à 3 chiffres après la virgule.

```
moyenne_2015 = round(gaz_2015/nb_habitants_2015 * 1000, 3)
moyenne_2020 = round(gaz_2020/nb_habitants_2020 * 1000, 3)
print(moyenne_2015) # 9.298
print(moyenne_2020) # 8.214
```

La moyenne pondérée des émissions de gaz à effet de serre de l'UE en 2020 est plus faible qu'en 2015. Les émissions de gaz à effet de serre par habitant dans l'UE ont donc tendance à diminuer.

Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

- On écrit le résultat dans un fichier txt.

Sur Spyder

```
moyenne_gaz = open("chemin\\Part1_Question4.txt", 'w')
phrase = ["phrase avec les résultats"]
moyenne_gaz.writelines(phrase)
```

Sur Pycharm

```
moyenne_gaz = open("part1question4.txt", "w")
moyenne_gaz = open("part1question4.txt", "r+")

moyenne_gaz.write(chaîne de caractères + str(moyenne_2015) + chaîne de
caractères + str(moyenne_2020) + chaîne de caractères)
```

5. Appartenance à l'une des 3 catégories en fonction de sa production de gaz à effet de serre

En premier lieu, il faut regrouper les données annuelles.

- Moyenne des 6 années pour chaque pays.

```
emission_moyenne = []
i = 0
while i <= len(Gas_effet_serre_annuel) - 7 :
    emission_moyenne.append(Gas_effet_serre_annuel[i])
    moyenne = sum(Gas_effet_serre_annuel[i+1:i+6+1]) / 6
    # on va jusqu'à i + 7 car même si on veut aller jusqu'à la 6e année (2020)
    # l'index ([a:b]) va de a jusqu'à b-1
    emission_moyenne.append(moyenne)
    i = i + 7
```

- On peut maintenant déterminer combien de pays figurent dans chaque catégorie.

```
nombre_pays_categ_1 = 0
nombre_pays_categ_2 = 0
nombre_pays_categ_3 = 0
i = 1
```

Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

```
while i <= len(emission_moyenne) - 1:
    if emission_moyenne[i] <= 100000 :
        nombre_pays_categ_1 += 1
        i = i + 2
    if emission_moyenne[i] <= 200000 :
        nombre_pays_categ_2 += 1
        i = i + 2
    else :
        nombre_pays_categ_3 += 1
        i = i + 2
```

- Dans quelle catégorie se trouve le plus de pays ?

```
valeurs = {"catégorie 1":nombre_pays_categ_1, "catégorie
2":nombre_pays_categ_2, "catégorie 3":nombre_pays_categ_3}
# on crée un dictionnaire pour plus de facilité

if max(valeurs.get("catégorie 1"), valeurs.get("catégorie 2"), valeurs.get("catégorie
3")) == valeurs.get("catégorie 1"):
    print("La catégorie contenant le plus de pays est la catégorie 1")
if max(valeurs.get("catégorie 1"), valeurs.get("catégorie 2"), valeurs.get("catégorie
3")) == valeurs.get("catégorie 2"):
    print("La catégorie contenant le plus de pays est la catégorie 2")
else :
    print("La catégorie contenant le plus de pays est la catégorie 3")
```

6. Bonus - Tableaux

Pour utiliser des tableaux, il nous faut la fonction *array* appartenant à la librairie *numpy*.

```
from numpy import array
```

- Tableau des deux pays produisant le plus d'émissions de gaz à effet de serre dans l'UE.

Puisque la liste *liste_max* est triée par ordre décroissant, pour avoir les deux maximums, on prend le nom du premier et deuxième pays (position 0 et 2) et la valeur qui les suit.

```
Tableau_emission_gaz_max = array([[liste_max[0], liste_max[2]], [liste_max[1],  
liste_max[3]]])
```

- Tableau des deux pays produisant le moins d'émissions de gaz à effet de serre dans l'UE.

Pour avoir le minimum, on prend les valeurs en partant de la fin.

Pour prendre la fin, on utilise un index négatif et celui-ci commence à -1 et non 0.

```
Tableau_emission_gaz_min = array([[liste_max[-2], liste_max[-4]], [liste_max[-1],  
liste_max[-3]]])
```

On obtient des tableaux de taille 2x2, avec en première ligne les noms des deux pays et en deuxième, leur émission.

PARTIE 2

Avant de commencer, on prépare les fichiers *PIB2015-2020* et *Recettes fiscales environnementales2015-2020* avec la même [méthode](#) que précédemment.

1. Conversion en pourcentage du PIB
 - On utilise en même temps les données des deux fichiers/listes.

```
Recettes_PIB = []
pays = 0
a = 1
while pays <= len(Recettes_fiscales_env) - 1 :
    Recettes_PIB.append(Recettes_fiscales_env[pays][0])
    # on ajoute le nom du pays
    while a <= 6 :
        pourcentage = (Recettes_fiscales_env[pays][a] / PIB[pays][a]) * 100
        Recettes_PIB.append(round(pourcentage,3))
        # on calcule le pourcentage et on l'ajoute dans la liste
        a = a + 1
    pays = pays + 1 # on passe au pays suivant
    a = 1
```

2. Trier les pays de l'UE en fonction des recettes fiscales environnementales (% du PIB) en 2020

Comme dans la [Partie 1](#).

3. Quel pays de l'UE a le moins et le plus de recettes fiscales environnementales en 2020 ?

Comme dans la [Partie 1](#).

4. Moyenne pondérée des recettes fiscales environnementales de l'UE en 2015 et 2020

Comme dans la [Partie 1](#).

La moyenne pondérée des émissions de gaz à effet de serre de l'UE en 2020 est plus faible qu'en 2015. Les émissions de gaz à effet de serre par habitant dans l'UE ont donc tendance à diminuer.

Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

5. Bonus - Graphiques

Pour ce faire, nous avons besoin d'importer la librairie *matplotlib.pyplot* pour utiliser les fonctions liées aux graphiques.

```
import matplotlib.pyplot as plt
```

- Graphique des deux pays ayant les plus grandes recettes fiscales environnementales (en % du PIB) et des deux pays ayant les plus petites recettes fiscales environnementales (en % du PIB).

Pour une bonne visibilité, nous avons choisi de faire un diagramme circulaire avec la fonction *plt.pie* de *matplotlib.pyplot*.

```
x = [liste_max[1], liste_max[3], liste_max[-1], liste_max[-3]]  
# x contient les valeurs de deux pays max et des deux pays min  
  
plt.figure(figsize = (8,8)) # permet de choisir la taille de la figure  
plt.pie(x, labels = [liste_max[0], liste_max[2], liste_max[-2], liste_max[-4]],  
        normalize = True, colors = ["#B0D8EA", "#C3E6C1", "#D3C3E4", "#F1D5EB"])  
# labels : permet d'associer un nom aux valeurs  
# il s'agit des noms des pays  
# normalize : fait un camembert complet en normalisant la somme des valeurs  
# colors : on choisit les couleurs pour chaque donnée  
plt.title("Diagramme des recettes maximums et minimums")  
# permet de donner un titre au graphique  
plt.show() # affiche le graphique
```

Diagramme circulaire

Diagramme des recettes maximums et minimums



PARTIE 3

1. Régression linéaire simple

Il faut d'abord avoir les moyennes annuelles (2 listes avec les moyennes de 6 années).

- Recettes_fiscales_env

Cette variable comporte autant de sous-listes qu'il n'y a de pays et chaque sous-liste comporte 7 éléments.

```
moyennes_recettes_annuelles = []
recettes_annuelles = 0
i = 0
j = 1
while j <= len(Recettes_fiscales_env[0]) - 1 : # <= 7 éléments - 1
    while i <= len(Recettes_fiscales_env) - 1 : # 27 pays - 1
        recettes_annuelles = recettes_annuelles + Recettes_fiscales_env[i][j]
        # dans le pays i, on prend la valeur de l'année j
        i = i + 1 # on passe au pays suivant
    moyennes_recettes_annuelles.append(recettes_annuelles /
len(Recettes_fiscales_env))
    # on en fait la moyenne annuelle est on l'ajoute dans la liste
    recettes_annuelles = 0
    i = 0
    j = j + 1 # on recommence pour le pays suivant
```

- Gas_effet_serre_annuel

Il s'agit d'une grande liste.

```
moyennes_emissions_annuelles = []
emissions_annuelles = 0
i = 1
j = 1
pays = 1
while j <= 6 : # inférieur au nombre d'années
    while pays <= 27 :
```

Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

```
emissions_annuelles = emissions_annuelles + Gas_effet_serre_annuel[j]
# émission de l'année j pour le pays pays
j = j + 7 # on passe à la même année, mais au pays suivant
pays = pays + 1
moyennes_emissions_annuelles.append(emissions_annuelles / 27)
recettes_annuelles = 0
i = i + 1 # on recommence pour l'année suivante
j = i
pays = 1
```

- Maintenant, nous pouvons faire la régression simple

```
x = moyennes_recettes_annuelles # variable explicative
y = moyennes_emissions_annuelles # variable à expliquer
```

- Graphique

On utilise les fonctions graphiques de la librairie `matplotlib.pyplot` que l'on avait déjà installé.

Pour faire un nuage de points, nous utilisons la fonction `scatter()`.

```
axes = plt.axes() # dessine la fenêtre graphique avec les axes
axes.grid() # dessine une grille pour une meilleure lisibilité du graphe
plt.scatter(y,x) # pour une meilleure lecture du graphique
plt.title("Nuage de point entre les recettes et les émissions")
plt.xlabel("Émissions annuelles moyennes")
plt.ylabel("Recettes fiscales environnementales moyennes")
plt.show()
```

On observe une augmentation des émissions en même temps que l'augmentation des recettes. Sauf pour 2020 qui est une valeur remarquable.

- Régression

Pour ce faire, nous avons besoin de la librairie `scipy` pour utiliser la fonction `linregress()`.

```
from scipy import stats
```

```
slope, intercept, r_value, p_value, std_err = stats.linregress(y, x)
# linregress() : renvoie plusieurs variables.
```

Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

```
# On s'intéresse particulièrement au slope et intercept  
reg = "y = " + str(intercept) + " + " + str(slope) + "x + u"
```

On obtient la régression suivante :

$$y = 11315.226467686532 + 0.0005628148595228931x + u$$

- Significativité

Nous voulons savoir si la variable "recettes environnementales moyennes"(x) explique la variable "émissions annuelles moyennes"(y).

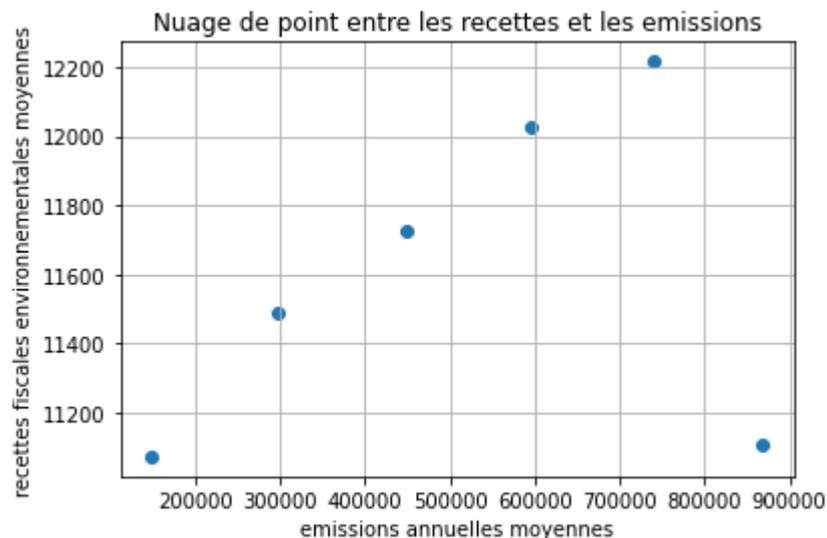
```
print(r_value) # 0.32391065858191
```

Le coefficient de détermination qui permet de vérifier la qualité du modèle est très faible (~32,4 %). Cela signifie que notre variable x explique peu la variable y.

Donc ce modèle ne semble pas significatif.

Seulement, il est important de noter qu'au vu de notre faible quantité de données (seulement 6), il est difficile d'apporter une conclusion définitive sur la pertinence du modèle.

Graphique de la régression



2. Bonus - Graphique Q-Q

- On importe plusieurs librairies qui nous seront utiles.

```
import numpy as np  
import pylab as py  
from numpy import array
```

Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

```
import statsmodels.api as sm # sur Pycharm
import scipy.stats as stats # sur Spyder
```

- On définit nos données.

Pour cela, on utilise les valeurs de 2017 dans la base *Gas_effet_serre_annuel*.

```
data_points = np.zeros(shape = 27)
# c'est un tableau d'une ligne contenant des 27 zéros
# les valeurs contenue sont par défaut des nombres à virgule
# on accède au tableau comme à une liste
i = 3 # la valeur de 2017 est à l'index 3 (4e position)
j = 0

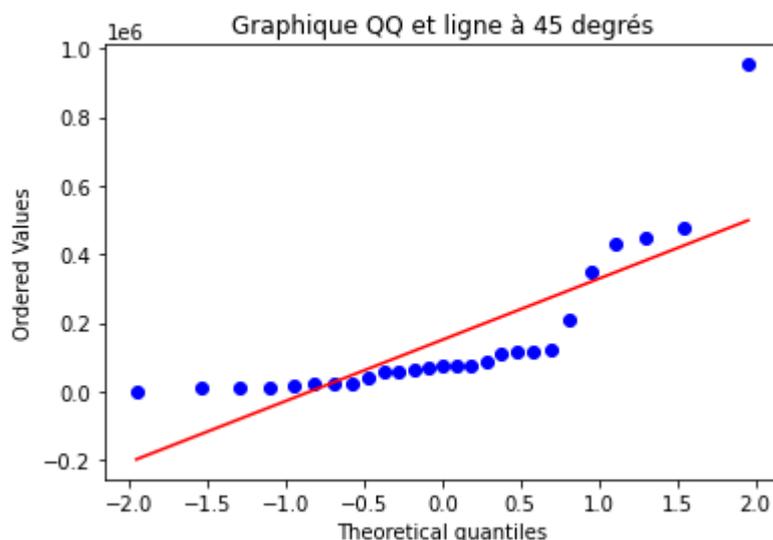
while i <= len(Gas_effet_serre_annuel) - 4 :
    # -4 est la position de la dernière valeur de 2017
    data_points[j] = Gas_effet_serre_annuel[i]
    # on remplace les 0 par les valeurs de 2017
    i = i + 7 # on passe à la valeur du pays suivant
    j = j + 1
```

- Graphique

Sur Spyder, on ne peut pas faire de *qq plot* puisque la librairie *statsmodels.api* n'y est pas disponible. Alors on réalise un *probability plot* et avec en paramètre une distribution normale, celui-ci qui est similaire au *qq plot*.

```
sm.qqplot(data_points, line='45') # sur Pycharm
stats.probplot(data_points, dist = "norm", plot = py) # sur Spyder
plt.title("Graphique QQ et ligne à 45 degrés")
py.show()
```

Graphique QQ

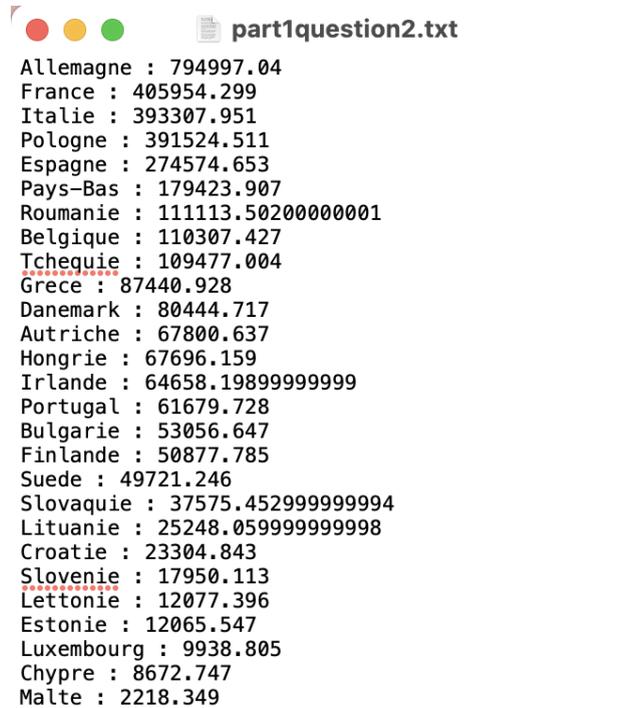


Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

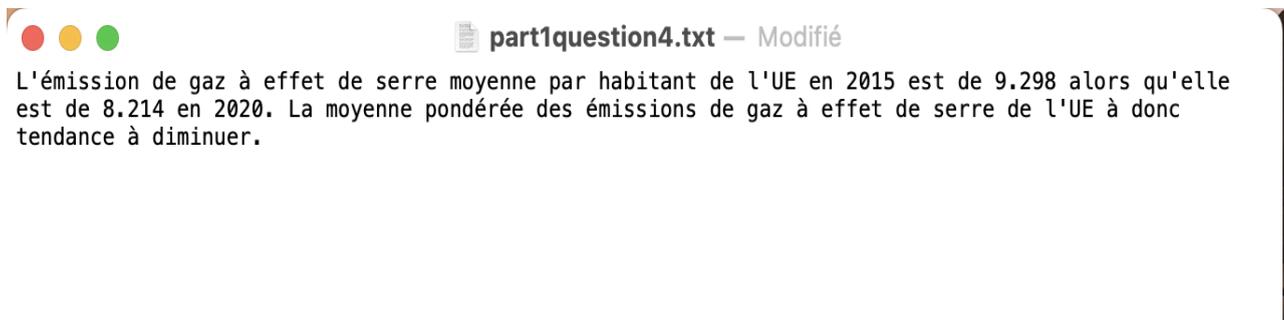
RÉSULTATS DANS LES FICHIERS TEXTE

Fichier txt "[part1question2](#)" triant dans l'ordre décroissant les émissions de GES des différents pays de l'UE en 2020 :



```
part1question2.txt
Allemagne : 794997.04
France : 405954.299
Italie : 393307.951
Pologne : 391524.511
Espagne : 274574.653
Pays-Bas : 179423.907
Roumanie : 111113.50200000001
Belgique : 110307.427
Tchequie : 109477.004
Grece : 87440.928
Danemark : 80444.717
Autriche : 67800.637
Hongrie : 67696.159
Irlande : 64658.19899999999
Portugal : 61679.728
Bulgarie : 53056.647
Finlande : 50877.785
Suede : 49721.246
Slovaquie : 37575.452999999994
Lituanie : 25248.059999999998
Croatie : 23304.843
Slovenie : 17950.113
Lettonie : 12077.396
Estonie : 12065.547
Luxembourg : 9938.805
Chypre : 8672.747
Malte : 2218.349
```

fichier txt "[part1question4](#)" :



```
part1question4.txt — Modifié
L'émission de gaz à effet de serre moyenne par habitant de l'UE en 2015 est de 9.298 alors qu'elle est de 8.214 en 2020. La moyenne pondérée des émissions de gaz à effet de serre de l'UE à donc tendance à diminuer.
```

Emma Vander Hoeden

Viviana Sanches Rodrigues Ribeiro

Fichier txt "[part2question2](#)" :

```
part2question2.txt
Grece : 3.766
Croatie : 3.28
Pays-Bas : 3.177
Danemark : 3.173
Bulgarie : 3.031
Italie : 3.023
Lettonie : 3.018
Slovenie : 2.941
Finlande : 2.725
Belgique : 2.54
Pologne : 2.534
Chypre : 2.437
Estonie : 2.393
Portugal : 2.377
Slovaquie : 2.346
Malte : 2.27
France : 2.172
Hongrie : 2.164
Autriche : 2.095
Suede : 2.001
Tchequie : 1.922
Lituanie : 1.918
Roumanie : 1.917
Espagne : 1.751
Allemagne : 1.689
Luxembourg : 1.379
Irlande : 1.212
```

Fichier txt "[part2question4](#)" :

```
part2question4.txt — Modifié
Les recettes fiscales environnementales moyennes par habitant de l'UE en 2015 sont de 691.384 euros alors qu'elles sont de 703.125 euros en 2020. La moyenne pondérée des recettes fiscales environnementales de l'UE à donc tendance à diminuer.
```