

# **LBIR1271**

## **Projet intégré en informatique et mathématiques appliquées**

### **Syllabus d'exercices 2021-2022**

Professeurs: Patrick Bogaert,  
Emmanuel Hanert,  
Marnik Vanclooster

Assistants : Antoine Saint-Amand  
Romain Duquenne  
François Toussaint

## A. Assigner et utiliser les variables

### A.1

Essayez de prédire le résultat des opérations suivantes :

- `(1+2) ** 3`
- `"Hey" * 4`
- `"Hey" + 3`
- `("So" + "da") * 2`
- `("Hey" * 4) / 2`
- `5 / 2`
- `5 // 2`
- `5 % 2`
- `str(4) * int("3")`
- `int("3") + float("3.2")`
- `str(3) * float("3.2")`
- `str(3/4) * 2`

Vérifiez vos réponses à l'aide de Python.

### A.2

Laquelle de ces instructions renverra un message d'erreur ? Que renvoient les autres ?

- `print(str(True) + "python")`
- `print("python" + True)`
- `print(False + True + 3)`
- `print(True + 0)`

Vérifiez vos réponses à l'aide de Python.

### A.3

Créez les variables `a`, `b` et `c` dont les valeurs respectives sont 2, 3 et 4.

Calculez ensuite la valeur de `d` à partir des trois variables définies, avec :

$$d = \frac{a - b^{c-2}}{1 - \frac{ab-2}{3} + c}$$

Affichez la valeur de `d` afin de vérifier qu'elle est bien égale à -1.909.

### A.4

Créez les variables `name` et `surname` contenant le prénom et le nom de Sarah Packman.

En utilisant ces deux variables, créez la variable `fullname` contenant son nom complet. Attention à bien insérer un espace entre son prénom et son nom !

Affichez ensuite la phrase :

*My sister's name is SARAH PACKMAN*

en utilisant votre variable `fullname`.

## B. Travailler avec des listes

### B.1

Quel(s) type(s) de données peut contenir une liste ? Choisissez parmi les propositions suivantes :

- Uniquement des *floats* et *integers*
- Uniquement des *floats*
- Uniquement des *strings*
- Tout type d'objet

### B.2

Quel sera l'output de ce code ?

```
x = [0, 2, -3]
x[1:] = [4, 6, 3]
print(x)
```

- [1, 4, 6, 3]
- [0, 4, 6, 3]
- [0, 6, 3]
- Message d'erreur

### B.4

Voici l'âge de dix jeunes (en années):

12 13 14 9 12 15 12 11 13 14

Créez une liste appelée `age` contenant les âges indiqués ci-dessus et affichez ensuite :

- Les âges du 3<sup>ème</sup> jusqu'au 6<sup>ème</sup> jeune ;
- Les âges du 6<sup>ème</sup> jusqu'au dernier jeune ;
- Les âges du 3<sup>ème</sup> jeune en partant de la fin de la liste jusqu'au dernier (en utilisant un index négatif)

Vous vous rendez compte que vous avez oublié un jeune de 10 ans. Ajoutez-le à la fin de votre liste.

Vous vous rendez aussi compte que le troisième jeune est en réalité un an plus jeune que ce que vous ne pensiez. Corrigez cela.

Calculez l'étendue de vos valeurs qui correspond à la différence entre la valeur maximale et minimale de vos données (utilisez des fonctions !). Nommez cette nouvelle variable `x` et affichez sa valeur.

Affichez tous les âges de votre liste par ordre croissant.

À l'aide d'une méthode du type `list`, trouvez le nombre de jeunes ayant 12 ans et affichez-le.

Affichez enfin les deux phrases suivantes :

*The oldest is X and the youngest is Y years old.*

en remplaçant X et Y par les bonnes valeurs de la liste. Attention, ceci doit se faire sous forme de code et il est donc exclu d'écrire les valeurs numériques directement dans votre phrase.

## C. Numpy

### C.1

Que va donner l'instruction `np.array([4,5]) < 6` ?

- Un *numpy array* de booléens
- Un *array* d'*integers*
- Une valeur booléenne

### C.2

Importez-le package `numpy` sous le nom `np`.

Créez la liste `radius` contenant les valeurs de rayons suivantes (en mètres) :

1.23 1.65 2.1 1.38 1.91

Affichez les rayons supérieurs à 1.5.

Créez la liste `area` contenant les aires de cercle que vous calculerez à partir des rayons contenus dans votre liste `radius`. Pour faire ce calcul, vous aurez besoin d'importer un autre package afin de pouvoir utiliser la valeur de  $\pi$ .

Affichez la phrase suivante en remplaçant l'étoile par la valeur adéquate :

*The area of circle n°2 is \* squared meters.*

Attention, ceci doit se faire sous forme de code et il est donc exclu d'écrire la valeur numérique directement dans votre phrase.

### C.3

Dans un circuit électrique à courant continu contenant une pile, une résistance et un condensateur chargé, si on déconnecte la pile on remarque que le courant ne chute pas à 0 instantanément mais diminue de manière exponentielle selon l'équation :

$$I(t) = I_0 e^{\frac{-t}{RC}}$$

où  $I(t)$  représente la valeur du courant (A),  $I_0$  l'intensité initiale du courant (A) avant déconnection de la pile,  $t$  le temps (s),  $R$  la résistance ( $\Omega$ ) et  $C$  la capacité (F).

Pour une intensité du courant qui vaut initialement 10 A, une résistance de 1000  $\Omega$  et une capacité de 50.10<sup>-6</sup> F, faites s'afficher la phrase suivante :

*L'intensité du courant 0.1 secondes après avoir déconnecté la pile vaut \* A.*

En remplaçant `*` par la valeur que vous aurez trouvée arrondie à deux décimales.

## D. Vos graphes avec matplotlib

### D.3

Vous disposez des données météorologiques de Louvain-La-Neuve et d'Ostende (1991-2020)<sup>1</sup> :

Mois	Températures (°C)	
	Louvain-La-Neuve	Ostende
Janvier	3.4	4.1
Février	3.8	4.4
Mars	6.6	6.5
Avril	9.7	9.2
Mai	13.5	12.6
Juin	16.4	15.4
Juillet	18.5	17.4
Août	18.1	17.6
Septembre	14.8	15
Octobre	11.1	11.4
Novembre	6.9	7.6
Décembre	4	4.8

Faites un graphe de type "*scatter plot*" présentant l'évolution de la température à Louvain-La-Neuve tout au long de l'année. Veillez à :

- Donner un titre aux axes ;
- Fixer les limites de l'axe des ordonnées pour qu'il soit compris entre 0 et 25°C ;
- Ce que les mois de l'année soient lisibles et ne se chevauchent pas les uns les autres.

Vous vous rendez compte que vous n'aimez pas les points bleus mais que vous préféreriez des croix noires. Faites le changement.

Vous souhaitez comparer les températures entre Louvain-La-Neuve et Ostende. Affichez-les sur un graphe. Veillez à :

- Utiliser des croix pour Louvain-La-Neuve et des losanges pour Ostende ;
- Utiliser du noir pour les croix et du rouge pour les losanges ;
- Afficher une légende sur votre graphe en haut à droite.

Pour faire plus "pro" :

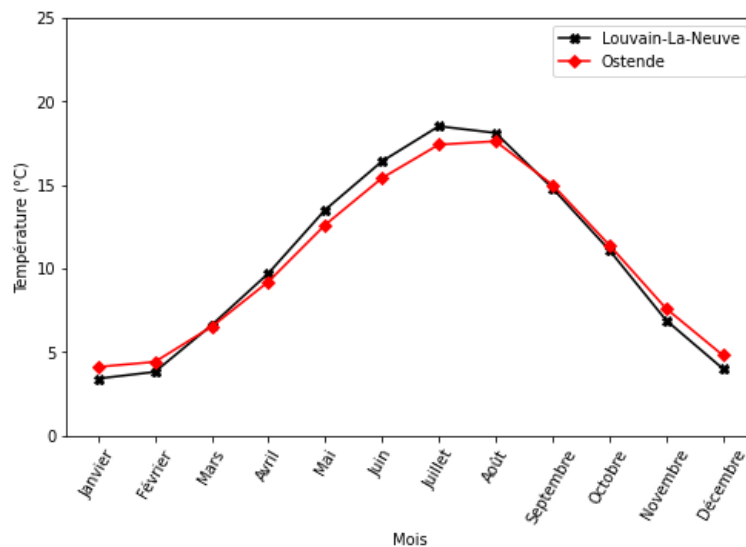
- Modifiez la taille du graphe (vous pouvez utiliser *figsize* pour cela et appliquer les dimensions [8.5] par exemple, ou en essayer d'autres) ;
- Ajoutez une courbe qui passe à travers vos points.

Trouver une commande qui permette de sauvegarder votre figure sur votre ordinateur au format .png (avec une ligne de code !) et faites-le.

*Remarque* : il se peut que cette manipulation rogne votre figure. Si c'est le cas, ajouter l'instruction `"bbox_inches='tight'"` dans la ligne de code qui sauvegarde votre figure.

<sup>1</sup> Institut Royal de Météorologie (s.d.). Le climat dans votre commune. <https://www.meteo.be/fr/climat/climat-de-la-belgique/climat-dans-votre-commune>. Consulté le 20/07/2021.

Voici ce à quoi vous devriez arriver :



## D.2

En physique, une oscillation harmonique simple peut être représentée mathématiquement par l'équation :

$$y(t) = A \sin(2\pi ft + \varphi)$$

où  $A$  représente l'amplitude du signal,  $f$  la fréquence (Hz),  $t$  le temps (s) et  $\varphi$  la constante de phase.

Sur un oscilloscope (cf. cours LBIR1221, "Ondes, optique et physique moderne"), il est possible d'avoir une représentation visuelle d'une telle onde, représentation pour laquelle l'axe des ordonnées représente alors une tension (V) (caractéristique lorsqu'on travaille avec un tel appareil).

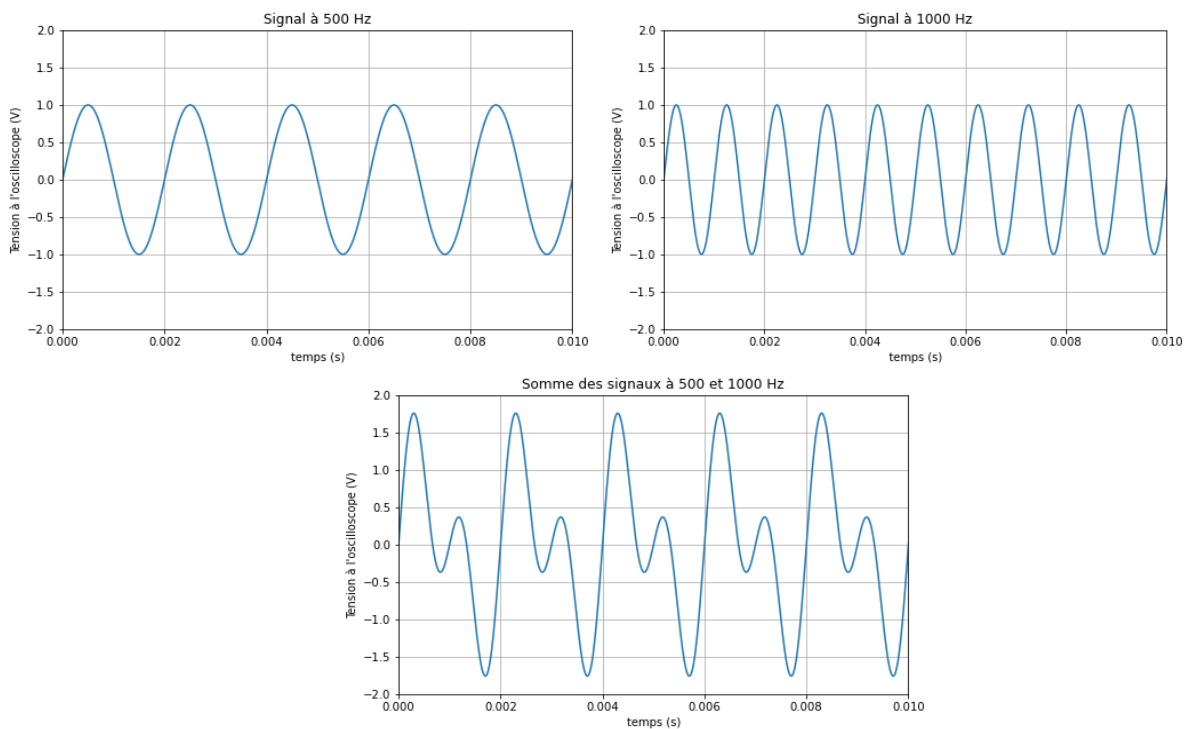
Pour une amplitude de 1 V, un déphasage nul et dans un laps de temps allant de 0 à 0.01 s, représentez graphiquement la propagation d'une onde acoustique reçue par l'oscilloscope et dont la fréquence est de 500 Hz. Veillez à :

- Réaliser un graphe du type "line plot" ;
- Ajouter un quadrillage ;
- Donner un titre aux axes ;
- Donner un titre au graphe "Signal à 500 Hz" ;
- Fixer les limites de l'axes des abscisses pour qu'il soit compris entre 0 et 0.01 s.

Vous vous demandez maintenant ce qu'afficherait l'oscilloscope s'il recevait en plus le signal d'une onde acoustique d'amplitude 1 V, de déphasage nul et dont la fréquence serait de 1000 Hz.

Représentez graphiquement la propagation de cette onde. Représentez ensuite la somme de cette nouvelle onde acoustique et de la première, dans le même laps de temps (0 à 0.01 s). En plus des spécificités du graphe précédent, veillez à ajouter un titre aux nouveaux graphes. Sauvegardez-les sur votre ordinateur.

Voici ce à quoi vous devriez arriver pour chaque graphe séparément :



### D.3

En reprenant l'équation de la décroissance du courant dans un circuit électrique à courant continu lorsqu'on déconnecte la pile :

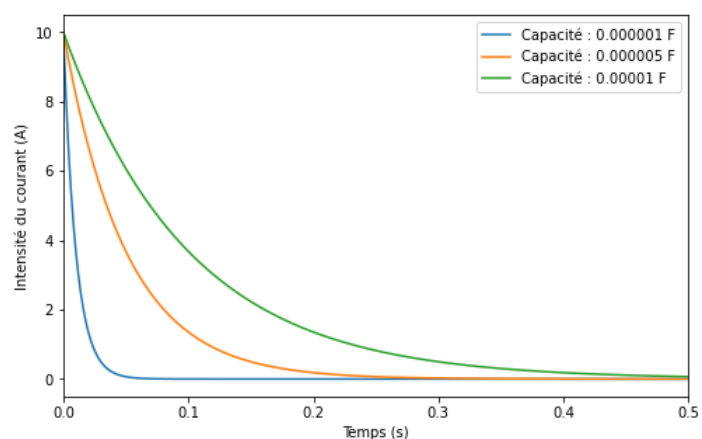
$$I(t) = I_0 e^{\frac{-t}{RC}}$$

Faites un graphe de l'évolution du courant de 0 à 0.5 s dans un circuit dont la résistance vaut 1000  $\Omega$  et l'intensité initiale du courant 10 A pour les valeurs de capacité suivantes :

- $C_1 = 10 \cdot 10^{-6}$  F ;
- $C_2 = 50 \cdot 10^{-6}$  F ;
- $C_3 = 100 \cdot 10^{-6}$  F.

Présentez ces trois courbes sur un seul et même graphe en veillant à afficher une légende et donner un titre aux axes. Que pouvez-vous dire à propos de l'influence de la capacité sur la décroissance du courant dans un circuit ?

Voici ce à quoi vous devriez arriver :



## E. Découvrir les dictionnaires et Pandas

### E.1

Encodez les valeurs suivantes sous forme d'un dictionnaire nommé *communes* :

Commune	Population
Anderlecht	118241
Etterbeek	47414
Forest	55746
Ixelles	86244
Evere	4394
Ottignies	30283

Afin de vous assurer que vous avez encodé les index correctement, affichez les index de votre variable *communes*. Vous devriez obtenir ceci :

```
dict_keys(['Anderlecht', 'Etterbeek', 'Forest', 'Ixelles', 'Evere',  
          'Ottignies'])
```

Affichez la phrase suivante en remplaçant l'étoile par la valeur adéquate :

*Il y a \* habitants à Ixelles.*

Attention, ceci doit se faire sous forme de code et il est donc exclu d'écrire la valeur numérique directement dans votre phrase.

Vous vous rendez compte que les données contiennent deux erreurs : la commune d'Evere contient en réalité 40394 habitants et la commune d'Ottignies ne se situant pas à Bruxelles, elle devrait être retirée de vos données. Faites ces modifications à l'aide de deux lignes de code et affichez ensuite votre dictionnaire *communes*.

### E.2

Chargez le package *pandas* sous le nom *pd*.

Le fichier '*emissions co2.csv*' contient les données d'émissions de CO<sub>2</sub> (tonnes/personne) (provenant de combustibles fossiles) de 178 pays entre 1970 et 2018. Chargez ces données sous forme de '*pandas dataframe*' en définissant comme index la première colonne.

Nous définissons les deux variables suivantes :

```
variable1 = emissions['2000']  
variable2 = emissions[['2000']]
```

Quelle est la différence entre ces deux variables ?

Affichez les émissions de CO<sub>2</sub> de la Belgique pour les années 1990, 2000 et 2010.

Affichez les émissions de CO<sub>2</sub> des 50<sup>e</sup>, 100<sup>e</sup> et 150<sup>e</sup> pays pour la première et la deuxième année enregistrées (1970 et 1971). Pour vérifier vos réponses, vous devriez obtenir les données de L'Egypte, de Malte et de Ste Lucie.

(Bonus) Affichez les émissions de CO<sub>2</sub> des 20 dernières années pour les 30 premiers pays du jeu de données.



## F. Boucles, opérateurs logiques et filtres

### F.1

Créez une liste `semaine` contenant les sept jours de la semaine. À l'aide d'une boucle, écrivez chaque jour de la semaine ainsi que les messages suivants :

- *Au travail* (s'il s'agit du lundi au jeudi) ;
- *Chouette c'est vendredi* (s'il s'agit du vendredi) ;
- *Repos ce weekend* (s'il s'agit du samedi ou du dimanche).

### F.2

Soit `impairs` la liste :

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21]
```

À partir de cette liste, créez une liste `pairs` dans laquelle tous les éléments sont incrémentés de 1.

### F.3

La liste ci-dessous représente la séquence d'un brin d'ADN :

```
["A", "C", "G", "T", "T", "A", "G", "C", "T", "A", "A", "C", "G"]
```

Créez un script qui transforme cette séquence en sa séquence complémentaire.

Rappel : la séquence complémentaire s'obtient en remplaçant A par T, T, par A, C par G et G par C.

### F.4

Créez un script qui dessine un triangle comme ceci :

```
*
**
***
****
*****
*****
*****
*****
*****
*****
```

### F.5

Vous êtes professeur et suite à un test vous disposez de la liste des points (/20) suivante :

```
[1, 12, 5, 12, 13, 15, 7, 18, 14, 9, 10, 9, 16]
```

Vous êtes également très (trop) gentil et vous souhaitez changer la note des étudiants étant en échec. À l'aide d'une boucle, passez en revue les éléments de votre liste et attribuez une note de 10 si elle était inférieure.

Vérifiez que votre opération a fonctionné en affichant la liste.

### Exercice 5

Chargez les données du fichier '`emissions_co2.csv`' sous forme de *pandas dataframe* en définissant comme index la première colonne. Nommez cette variable `emissions`. On s'intéresse aux émissions de CO2 de la Belgique en 2018. Créez la structure conditionnelle (`if...elif...else`) suivante :

- Si cette valeur est supérieure à 10, affichez la phrase : *La Belgique émet beaucoup de CO<sub>2</sub>* ;
- Si cette valeur est entre 5 et 10, affichez la phrase : *Les émissions de CO<sub>2</sub> de la Belgique sont moyennes* ;
- Si cette valeur est inférieure à 5, affichez la phrase : *La Belgique émet moins de CO<sub>2</sub> que la moyenne mondiale*.

Affichez toutes les valeurs d'émissions pour les pays dont les émissions de CO<sub>2</sub> en 2018 ont été supérieures à 20 tonnes/personne ou inférieures à 0.05 tonnes/personne. Vous devriez avoir sélectionné 7 pays.

(Exercice avancé) : Refaites le point précédent mais en affichant uniquement les données des 5 dernières années triées par ordre croissant sur base des valeurs de l'année 2018.

## F.6

À partir du `dataframe` `emissions` et à l'aide d'une boucle `for`, affichez pour chaque année la phrase suivante :

*La Belgique a émis \* tonnes de CO<sub>2</sub> par personne en \*\*.*

En remplaçant `*` par la valeur des émissions et `**` par l'année.

Nous aimerions ajouter une colonne (`MAX`) contenant la valeur du pic d'émissions de chaque pays depuis 1970. Cependant, une erreur s'est introduite dans les données et toutes les valeurs d'émissions de CO<sub>2</sub> supérieure à 50 pour l'année 1970 devraient être corrigées. Pour cela, faites une boucle `for` dans laquelle vous allez pour chaque pays :

- Vérifier si les émissions pour 1970 sont supérieures à 50 et si c'est le cas modifier cette valeur dans le `dataframe` `emissions` afin qu'elle soit égale à 50 ;
- Définir la valeur de la colonne `MAX` comme étant la valeur maximale d'émissions de ce pays depuis 1970.

## G. Écrire une fonction

### G.1

Créez une fonction qui renvoie le carré d'un argument. Elle doit afficher la phrase suivante :

*Le cube de la valeur \* est \*\*.*

En remplaçant \* par l'argument entré dans la fonction et \*\* par le résultat.

### G.2

Pour rappel, les oscillations harmoniques simples peuvent être représentées mathématiquement par l'équation :

$$y(t) = A \sin(2\pi ft + \varphi)$$

Où  $A$  représente l'amplitude du signal,  $f$  la fréquence (Hz),  $t$  le temps (s) et  $\varphi$  la constante de phase.

Créez une fonction `wave` prenant trois paramètres en entrée ( $A$ ,  $f$  et  $\varphi$ ) et qui affiche le même graphe que ce qui a été fait à l'exercice X.

*Remarque :* créez dans un premier temps la variable `t` qui comprendra des valeurs de temps comprises entre 0 et 0.01 s.

Testez votre fonction pour différentes valeurs d'amplitude, fréquence et déphasage.

### G.3

Nous aimerions créer deux fonctions : l'une nous donnera un résumé statistique d'une liste, l'autre nous permettra de centrer et réduire cette liste.

Créez une fonction nommée `stats` contenant un seul argument `mylist` et vous permettant :

- D'afficher un histogramme de `mylist` contenant 9 'bins' ;
- D'affichez les données suivantes :

Min: \*  
Max: \*  
Length: \*  
Range: \*  
Mean: \*

En remplaçant les \* par les valeurs adéquates. Pour la moyenne, aidez-vous de la fonction `mean` de `numpy`.

Créez une deuxième fonction nommée `norm` contenant l'argument `mylist` ainsi que l'argument `show` égal à `False` par défaut. Dans cette fonction :

Définissez la variable `newlist` qui correspond à votre liste `mylist` centrée et réduite.

Pour rappel :

$$Z = \frac{X - \mu}{\sigma}$$

où  $Z \sim N(0,1)$ , la variable  $X$  centrée et réduite. Approximez  $\mu$  et  $\sigma$  par leurs estimateurs usuels. Pour calculer l'écart-type de `mylist`, vous aurez besoin de la fonction `std` de `numpy`.

Créez une structure conditionnelle (`if`) vous permettant d'afficher un histogramme (contenant 9 'bins') de `newlist` si la variable `show` est égale à `True`.

Créez une liste contenant au moins 20 nombres au hasard et essayez vos deux fonctions `stats` et `norm` (n'oubliez pas d'essayer votre fonction `norm` avec l'argument `show=True`).

## H. Manipuler des données avec Pandas

Pour les exercices 1 à 3, vous aurez besoin du fichier `covid_vaccination.csv`. Chargez-le sous le nom `vaccin` en définissant la colonne `country` comme index. Ces données contiennent les variables suivantes au 18 janvier 2021 :

- `country` : le nom du pays (utilisez cette colonne comme index lorsque vous chargez les données);
- `continent` : le nom du continent sur lequel se trouve le pays;
- `total_vaccinations` : le nombre de personnes vaccinées contre la covid dans ce pays;
- `total_vaccinations_per_hundred` : le pourcentage de personnes vaccinées (covid) dans ce pays;
- `vaccines` : le(s) vaccin(s) utilisé(s) par ce pays.

Pour l'exercice 5, vous aurez besoin des données `emissions_co2.csv` que vous connaissez déjà. Chargez-les sous le nom `emissions` en définissant la première colonne comme index.

### H.1

Affichez un résumé de votre dataframe `vaccin`.

Affichez les lignes des pays ayant atteint un taux de vaccination supérieur à 5% en utilisant uniquement le vaccin "Pfizer/BioNTech" (indice : il y en a deux).

Vous disposez du pourcentage de personnes vaccinées ainsi que du nombre de personnes vaccinées, vous pouvez donc estimer la population de chaque pays grâce à ces deux colonnes. Ajoutez une colonne "`population`" à votre dataframe contenant les populations estimées des pays.

Affichez ensuite les lignes des 5 pays les plus peuplées.

Au point précédent, vous avez dû vous rendre compte que la population estimée d'un certain pays est aberrante. Remplacez cette valeur aberrante par la valeur correcte que vous trouverez sur internet.

### H.2

Affichez le nombre de pays par continent.

On aimerait connaître la répartition du nombre de personnes vaccinées parmi les continents. Pour cela, commencez par définir `total_vaccinated` qui correspond au nombre total de personnes vaccinées dans le monde.

Créez aussi `total_vaccinated_by_cont`, le nombre de personnes vaccinées par continent (à l'aide de `groupby()`).

Affichez ensuite le rapport de ces deux variables pour obtenir la répartition demandée (indice : pour vérifier votre réponse, 61.45% des personnes vaccinées dans le monde vivent en Europe).

Affichez le nombre de personnes vaccinées par continent en utilisant cette fois-ci la méthode `.pivot_table()`.

### H.3

Dans cet exercice, nous allons nous intéresser aux valeurs manquantes dans nos données. On aimerait connaître le pourcentage de lignes ne contenant pas de données sur le nombre de personnes vaccinées. Pour cela, définissez `nodata` et `Ndata` correspondant respectivement au nombre de valeurs manquantes dans la colonne "`total_vaccinations`" et au nombre de lignes dans le dataframe (indice : utilisez l'attribut `shape` pour trouver `Ndata`).

Affichez ensuite la phrase suivante :

*">% des pays recensés ont des données incomplètes*

En remplaçant le \* par la valeur adéquate.

Définissez `vaccin_complete` contenant toutes les lignes complètes de votre dataframe `vaccin`.

Sauvez ensuite `vaccin_complete` au format `.csv` sous le nom `covid_vaccination_complete.csv`

#### H.4

Définissez le dataframe `results` contenant les résultats de plusieurs étudiant·e·s au test 1 et au test 2:

Name	Test 1	Test 2
Alice	8	12
Romain	12	14
June	17	14
Baptiste	10	8
Laure	15	16

Définissez ensuite la première colonne comme index de votre dataframe à l'aide de la méthode `.set_index()`. Affichez `results` afin de vérifier que vous avez fait l'exercice correctement.

#### H.5

Créez et affichez la figure ci-dessous contenant l'histogramme des valeurs d'émissions de CO2 pour l'année 1970 et pour l'année 2010 (utilisez un indice de transparence de 0.7). N'oubliez pas de préciser la légende des couleurs ainsi que le titre des axes et de votre graphique.

