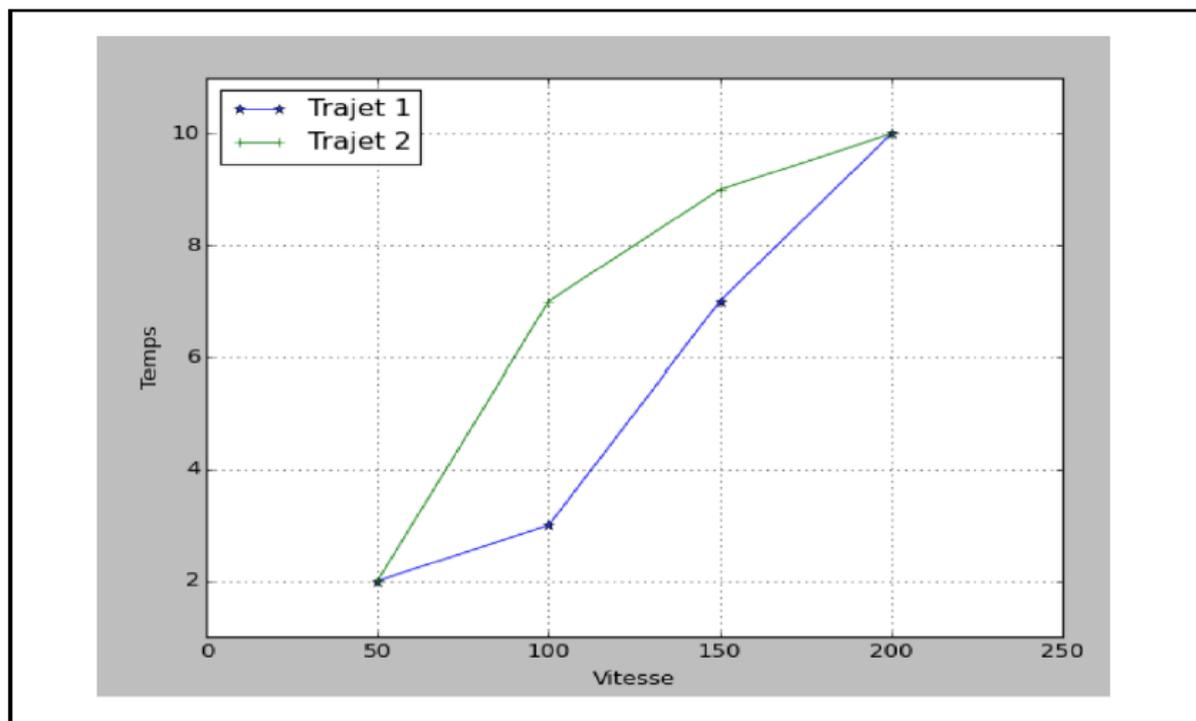


Utilisation de modules et bibliothèques

Exercice n°1 : Premiers graphiques avec la bibliothèque Matplotlib

- Écrire un script Python conduisant au graphique suivant :



L'affichage de la légende s'effectue à l'aide de la commande :

```
plt.legend(loc='upper left')
```

- On définit la fonction f sur \mathbb{R} par :

$$f(x) = \begin{cases} x^2 + 1 & \text{pour } x > 2 \\ -x^2 - 1 & \text{pour } x \leq 2 \end{cases}$$

Tracer la courbe représentative de f sur $[-6; 7]$ avec $N = 500$ points.

- On considère g définie sur \mathbb{R} par :

$$g(x) = \sqrt[3]{x}$$

Tracer la courbe représentative de g sur $[-7; 10]$ avec $N = 500$ points.

Construire le graphique sur $[-1; 3]$ de la fonction h définie sur \mathbb{R} par :

$$h(x) = \sqrt[3]{(e^x - 7)^2 (1 - 2x)}$$

- On définit la fonction k sur \mathbb{R} , 2π -périodique et paire, par :

$$k(x) = x \text{ sur } [0; \pi]$$

Tracer le graphique de k sur $[-10\pi; 10\pi]$ avec $N = 500$ points.

Exercice n°2 : Création et utilisation d'un module

Dans un premier temps, nous allons créer un module nommé `stats` qui contiendra trois fonctions :

- Une fonction `somme` qui prend une liste `L` non vide en argument et renvoie la somme des éléments qu'elle contient
- Une fonction `moyenne` qui prend une liste `L` non vide en argument et renvoie la moyenne des éléments qu'elle contient
- Une fonction `moyenne_quad` qui prend une liste `L` non vide en argument et renvoie la moyenne des carrés des éléments qu'elle contient

Ce module pourra par la suite être complété par d'autres fonctions.

- Dans un fichier `stats.py`, écrire les fonctions `somme`, `moyenne` et `moyenne_quad` (l'utilisation de fonctions déjà présentes dans python est interdite)
- Après avoir fermé le fichier `stats.py`, sans changer de répertoire de travail, dans le shell, importer le module `stats` sous le nom `st` puis tester les trois fonctions.

Nous allons maintenant utiliser les fonctions de notre module afin d'effectuer deux types de calcul statistique :

- Une série statistique est donnée par une liste dont les éléments sont des listes de deux nombres représentant une valeur et un effectif. À l'aide des fonctions du module `stats`, écrire dans un nouveau fichier une fonction `moyenne_pond` qui prend en argument une liste représentant une série statistique et renvoie la moyenne pondérée de cette série.
- À l'aide des fonctions du module `stats`, écrire une fonction `variance` qui prend en paramètre une liste de nombres et renvoie la variance de ces nombres. Pour rappel, la calcul de la variance s'effectue ainsi : on calcule la somme des carrés des valeurs, somme que l'on divise par le nombre de valeurs, autrement dit on calcule la moyenne des carrés des valeurs. Ensuite, on retranche à cette moyenne le carré de la moyenne des valeurs.

Exercice n°3 : Utilisation de fichiers externes

Lorsqu'un fichier est appelé ou est créé par un programme, celui-ci doit savoir où aller le chercher et où l'enregistrer. Ce paramétrage s'effectue dans le menu « File Browser » de Pizo.

Pour les manipulations qui vont suivre nous allons choisir de mettre les fichiers manipulés et le fichier contenant les programmes Python dans le même répertoire.

Pour cela :

- Il faut créer un répertoire de travail dans notre espace personnel : créez par exemple un répertoire « python » dans votre espace personnel
- Il faut préciser dans les préférences de Pizo que désormais, le répertoire de travail global est « python ». Pour cela, se rendre dans le menu « File Browser », rechercher le répertoire désiré pour stocker le travail puis demander « Go To This Directory In The Current Shell ».
- Il faut sauvegarder les fichiers textes que l'on souhaite lire dans ce répertoire.

1. Lecture de fichier texte

- À l'aide d'un éditeur de texte, créer un fichier contenant les différents paragraphes :

Toto va dans la cuisine
Titi va se faire manger
Tutu aime la danse

L'enregistrer sous le nom « toto.txt » dans le répertoire « python » précédent.

- Sous python, ouvrir en mode lecture le fichier toto.txt dans une variable toto. À l'aide de la fonction `readline`, stocker chaque ligne dans une variable de type `str` (les trois variables pourront être notées `C1`, `C2`, `C3`). Refermer le fichier une fois l'opération effectuée.
- Afficher `C1`, `C2`, `C3`.
- Ouvrir à nouveau en mode lecture le fichier toto.txt dans la variable toto. À l'aide de la fonction `readlines`, stocker le contenu du fichier dans une variable `C` de type `str`. Refermer le fichier une fois l'opération effectuée.
- Afficher `C`. Que remarque-t-on ? À quoi correspond la codification `\n` ? Comment peut-on supprimer ces éléments afin de retrouver le texte initial ? Comment appelle-t-on les différents paragraphes du texte ?

2. Lecture d'un fichier de type Excel

Pour convertir un fichier excel en un fichier de données lisible par python, le fichier doit être enregistré au format *.csv.

- Créer un fichier excel ou openoffice contenant la ligne les données suivantes :

toto	23	rouge
titi	12	vert

tutu	57	rose
------	----	------

- Enregistrer ce fichier au format csv sous le nom : toto.csv
- Ouvrir ce fichier à l'aide du blocnote pour voir ce qu'il contient.
- Concevoir un programme python permettant de calculer la somme des nombres de la deuxième colonne.

3. Création d'un fichier texte / Ajout dans un fichier existant

Le mode d'ouverture d'un fichier est différent suivant que celui existe déjà ou doit être créer (ou que l'on souhaite écraser la version précédente).

- Créer un fichier en écriture nommé «resultats.txt» et dont la copie python est nommés s. Vérifiez que ce fichier a bien été créer dans votre répertoire de travail global et vérifiez son contenu.
- Y écrire : « Voyons si cela marche. »
- Fermer le fichier puis vérifier que le contenu a été modifié.
- Ouvrir à nouveau le fichier «resultats.txt» en mode ajout. Y écrire : J'ajoute une phrase.
- Fermer le fichier puis vérifier que le contenu a été modifié.