TP10_2024_sujet

March 14, 2024

1 Calcul Scientifique – L1

2 TP Noté – durée 2h

Travail réalisé par : NOM / PRENOM

3 Instructions

1. Ce TP doit être fait sur les machines de l'université, individuellement.
2. Les accès à internet sont interdits à l'exception de ecampus (votre journal d'accès internet durant l'épreuve sera conservé)
3. Vous pouvez accéder à vos TP sur votre espace personnel.
4. Les téléphones portables doivent être éteints au fond des sacs
5. Il est interdit de communiquer.
6. Vous devez rendre dans la zone prévue pour votre salle votre fichier ipynb qui doit porter votre nom en respectant l'heure de fin prévue.
7. Les exercices sont indépendants

3.1 Partie I : numpy – matplotlib

Résoudre les questions de cette partie à l'aide de fonctions de la librairie `numpy` sans utiliser ni boucle `for` ou `while` ni d'instructions `if`.

3.1.1 Exercice 1 :

[1]:
```
import numpy as np
```

Le tableau qui suit contient des mesures faites sur des fleurs. Chaque ligne du tableau contient des dimensions pour une fleur. Les colonnes, au nombre de 5, contiennent respectivement :

1. longueur du sépale en cm
2. largeur du sépale en cm
3. longueur du pétale en cm
4. largeur du pétale en cm
5. le type qui est un chiffre parmi 0,1,2 qui indique le nom de la fleur.

[2]:

```
iris = [[5.1,3.5,1.4,0.2,0],[4.9,3.0,1.4,0.2,0],[4.7,3.2,1.3,0.2,0],[4.6,3.1,1.
↪5,0.2,0],[5.0,3.6,1.4,0.2,0],[5.4,3.9,1.7,0.4,0],[4.6,3.4,1.4,0.3,0],[5.0,3.
↪4,1.5,0.2,0],[4.4,2.9,1.4,0.2,0],[4.9,3.1,1.5,0.1,0],[5.4,3.7,1.5,0.2,0],[4.
↪8,3.4,1.6,0.2,0],[4.8,3.0,1.4,0.1,0],[4.3,3.0,1.1,0.1,0],[5.8,4.0,1.2,0.
↪2,0],[5.7,4.4,1.5,0.4,0],[5.4,3.9,1.3,0.4,0],[5.1,3.5,1.4,0.3,0],[5.7,3.8,1.
↪7,0.3,0],[5.1,3.8,1.5,0.3,0],[5.4,3.4,1.7,0.2,0],[5.1,3.7,1.5,0.4,0],[4.6,3.
↪6,1.0,0.2,0],[5.1,3.3,1.7,0.5,0],[4.8,3.4,1.9,0.2,0],[5.0,3.0,1.6,0.2,0],[5.
↪0,3.4,1.6,0.4,0],[5.2,3.5,1.5,0.2,0],[5.2,3.4,1.4,0.2,0],[4.7,3.2,1.6,0.
↪2,0],[4.8,3.1,1.6,0.2,0],[5.4,3.4,1.5,0.4,0],[5.2,4.1,1.5,0.1,0],[5.5,4.2,1.
↪4,0.2,0],[4.9,3.1,1.5,0.1,0],[5.0,3.2,1.2,0.2,0],[5.5,3.5,1.3,0.2,0],[4.9,3.
↪1,1.5,0.1,0],[4.4,3.0,1.3,0.2,0],[5.1,3.4,1.5,0.2,0],[5.0,3.5,1.3,0.3,0],[4.
↪5,2.3,1.3,0.3,0],[4.4,3.2,1.3,0.2,0],[5.0,3.5,1.6,0.6,0],[5.1,3.8,1.9,0.
↪4,0],[4.8,3.0,1.4,0.3,0],[5.1,3.8,1.6,0.2,0],[4.6,3.2,1.4,0.2,0],[5.3,3.7,1.
↪5,0.2,0],[5.0,3.3,1.4,0.2,0],[7.0,3.2,4.7,1.4,1],[6.4,3.2,4.5,1.5,1],[6.9,3.
↪1,4.9,1.5,1],[5.5,2.3,4.0,1.3,1],[6.5,2.8,4.6,1.5,1],[5.7,2.8,4.5,1.3,1],[6.
↪3,3.3,4.7,1.6,1],[4.9,2.4,3.3,1.0,1],[6.6,2.9,4.6,1.3,1],[5.2,2.7,3.9,1.
↪4,1],[5.0,2.0,3.5,1.0,1],[5.9,3.0,4.2,1.5,1],[6.0,2.2,4.0,1.0,1],[6.1,2.9,4.
↪7,1.4,1],[5.6,2.9,3.6,1.3,1],[6.7,3.1,4.4,1.4,1],[5.6,3.0,4.5,1.5,1],[5.8,2.
↪7,4.1,1.0,1],[6.2,2.2,4.5,1.5,1],[5.6,2.5,3.9,1.1,1],[5.9,3.2,4.8,1.8,1],[6.
↪1,2.8,4.0,1.3,1],[6.3,2.5,4.9,1.5,1],[6.1,2.8,4.7,1.2,1],[6.4,2.9,4.3,1.
↪3,1],[6.6,3.0,4.4,1.4,1],[6.8,2.8,4.8,1.4,1],[6.7,3.0,5.0,1.7,1],[6.0,2.9,4.
↪5,1.5,1],[5.7,2.6,3.5,1.0,1],[5.5,2.4,3.8,1.1,1],[5.5,2.4,3.7,1.0,1],[5.8,2.
↪7,3.9,1.2,1],[6.0,2.7,5.1,1.6,1],[5.4,3.0,4.5,1.5,1],[6.0,3.4,4.5,1.6,1],[6.
↪7,3.1,4.7,1.5,1],[6.3,2.3,4.4,1.3,1],[5.6,3.0,4.1,1.3,1],[5.5,2.5,4.0,1.
↪3,1],[5.5,2.6,4.4,1.2,1],[6.1,3.0,4.6,1.4,1],[5.8,2.6,4.0,1.2,1],[5.0,2.3,3.
↪3,1.0,1],[5.6,2.7,4.2,1.3,1],[5.7,3.0,4.2,1.2,1],[5.7,2.9,4.2,1.3,1],[6.2,2.
↪9,4.3,1.3,1],[5.1,2.5,3.0,1.1,1],[5.7,2.8,4.1,1.3,1],[6.3,3.3,6.0,2.5,2],[5.
↪8,2.7,5.1,1.9,2],[7.1,3.0,5.9,2.1,2],[6.3,2.9,5.6,1.8,2],[6.5,3.0,5.8,2.
↪2,2],[7.6,3.0,6.6,2.1,2],[4.9,2.5,4.5,1.7,2],[7.3,2.9,6.3,1.8,2],[6.7,2.5,5.
↪8,1.8,2],[7.2,3.6,6.1,2.5,2],[6.5,3.2,5.1,2.0,2],[6.4,2.7,5.3,1.9,2],[6.8,3.
↪0,5.5,2.1,2],[5.7,2.5,5.0,2.0,2],[5.8,2.8,5.1,2.4,2],[6.4,3.2,5.3,2.3,2],[6.
↪5,3.0,5.5,1.8,2],[7.7,3.8,6.7,2.2,2],[7.7,2.6,6.9,2.3,2],[6.0,2.2,5.0,1.
↪5,2],[6.9,3.2,5.7,2.3,2],[5.6,2.8,4.9,2.0,2],[7.7,2.8,6.7,2.0,2],[6.3,2.7,4.
↪9,1.8,2],[6.7,3.3,5.7,2.1,2],[7.2,3.2,6.0,1.8,2],[6.2,2.8,4.8,1.8,2],[6.1,3.
↪0,4.9,1.8,2],[6.4,2.8,5.6,2.1,2],[7.2,3.0,5.8,1.6,2],[7.4,2.8,6.1,1.9,2],[7.
↪9,3.8,6.4,2.0,2],[6.4,2.8,5.6,2.2,2],[6.3,2.8,5.1,1.5,2],[6.1,2.6,5.6,1.
↪4,2],[7.7,3.0,6.1,2.3,2],[6.3,3.4,5.6,2.4,2],[6.4,3.1,5.5,1.8,2],[6.0,3.0,4.
↪8,1.8,2],[6.9,3.1,5.4,2.1,2],[6.7,3.1,5.6,2.4,2],[6.9,3.1,5.1,2.3,2],[5.8,2.
↪7,5.1,1.9,2],[6.8,3.2,5.9,2.3,2],[6.7,3.3,5.7,2.5,2],[6.7,3.0,5.2,2.3,2],[6.
↪3,2.5,5.0,1.9,2],[6.5,3.0,5.2,2.0,2],[6.2,3.4,5.4,2.3,2],[5.9,3.0,5.1,1.8,2]]
```

[3]:
```
# à compléter
```

1) Après avoir converti le tableau en ‘array’ numpy, déterminer et afficher le nombre de fleurs total (nombre de lignes du tableau) et le nombre de fleurs de chacune des trois types.

[4]:
```
# à compléter
```

```
Nombre de fleurs: 150
Nombre de fleurs du type  0 : 50
Nombre de fleurs du type  1 : 50
Nombre de fleurs du type  2 : 50
```

2) Calculer pour l'ensemble des fleurs de type 2 les valeurs moyennes des 4 mesures (longueur sépale, largeur sépale, longueur pétale, largeur pétale).

[5]:
```python
# à compléter
```

```
Valeurs moyennes des mesures, classe  2  : [6.588 2.974 5.552 2.026]
```

3) Calculer les valeurs moyennes sur l'ensemble des classes des 4 mesures (longueur sépales, largeur sépales, longueur pétales, largeur pétales).

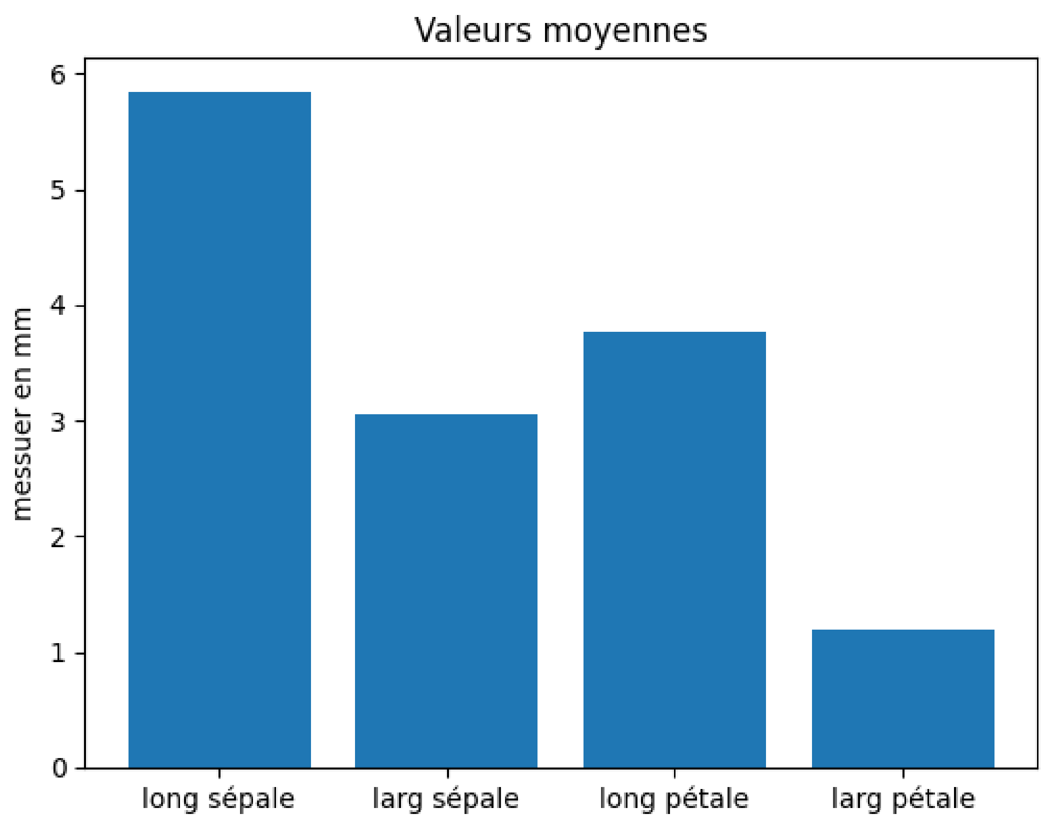
Visualiser ces résultats comme sur le graphique ci-dessous (barres verticales)

[6]:
```python
# à compléter

import matplotlib.pyplot as plt
%matplotlib inline

# à compléter
```

```
Valeurs moyennes des mesures : [5.84333333 3.054      3.75866667 1.19866667]
```



4) On considère les 4 tableaux relévés suivants:

longs: qui contient les longueurs de sépales pour l'ensemble des fleurs

largs: qui contient les largeurs de sépales pour l'ensemble des fleurs

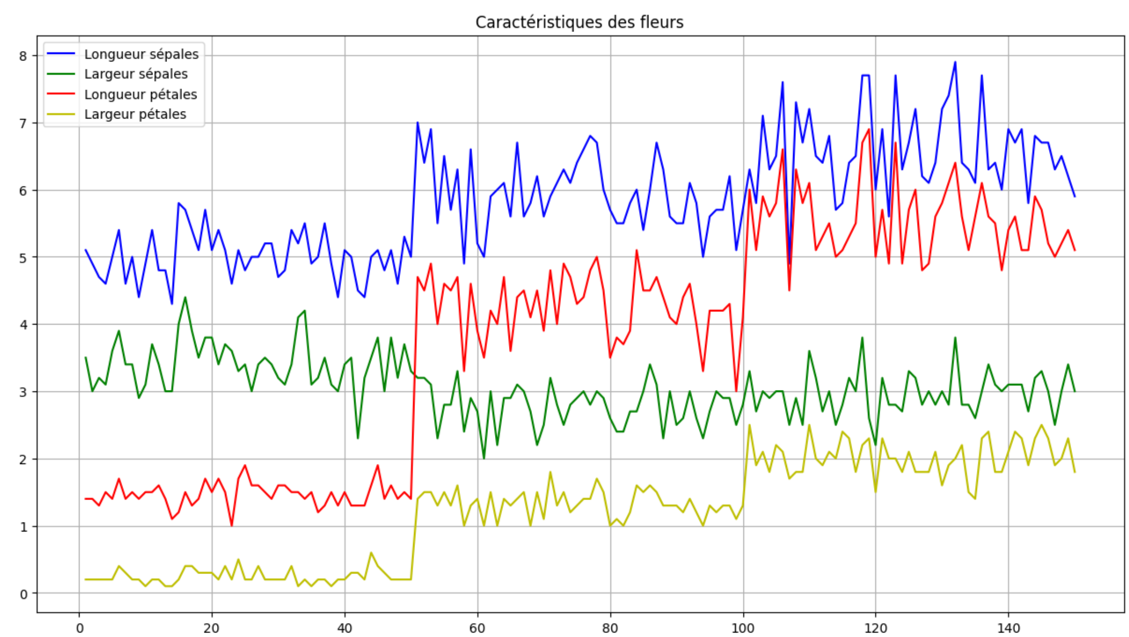
longp: qui contient les longueurs de pétales pour l'ensemble des fleurs

largp: qui contient les largeurs de pétales pour l'ensemble des fleurs

Calculer ces 4 tableaux.

Faire un schéma pour ces 4 relevés en fonction du numéro de la fleur(à partir de 1). Le schéma doit etre semblable à celui donné ci-dessous (légende, grille, couleurs , points).

[7]:
```
import matplotlib.pyplot as plt
%matplotlib inline
# à compléter
```

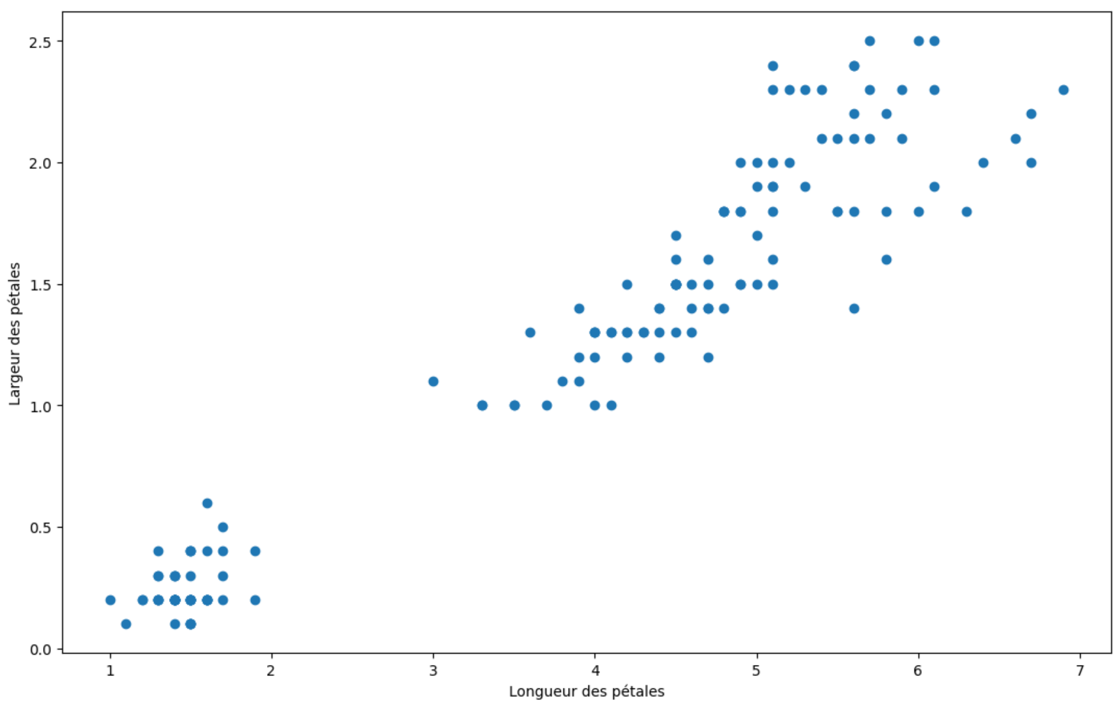


5) On reprend les deux tableaux longp et largp qui regroupent respectivement toutes les longueurs des pétales (colonne2) et toutes les largeurs des pétales (colonne 3)

Tracer un graphique donnant la largeur des pétales (colonne 3) en fonction de la longueur des pétales (colonne 2), pour l'ensemble des 150 fleurs.

[8]:
```
# à compléter
```

```
largp [0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 0.2 0.2 0.1 0.1 0.2 0.4 0.4 0.3
 0.3 0.3 0.2 0.4 0.2 0.5 0.2 0.2 0.4 0.2 0.2 0.2 0.2 0.4 0.1 0.2 0.1 0.2
 0.2 0.1 0.2 0.2 0.3 0.3 0.2 0.6 0.4 0.3 0.2 0.2 0.2 0.2 1.4 1.5 1.5 1.3
 1.5 1.3 1.6 1.  1.3 1.4 1.  1.5 1.  1.4 1.3 1.4 1.5 1.  1.5 1.1 1.8 1.3
 1.5 1.2 1.3 1.4 1.4 1.7 1.5 1.  1.1 1.  1.2 1.6 1.5 1.6 1.5 1.3 1.3 1.3
 1.2 1.4 1.2 1.  1.3 1.2 1.3 1.3 1.1 1.3 2.5 1.9 2.1 1.8 2.2 2.1 1.7 1.8
 1.8 2.5 2.  1.9 2.1 2.  2.4 2.3 1.8 2.2 2.3 1.5 2.3 2.  2.  1.8 2.1 1.8
 1.8 1.8 2.1 1.6 1.9 2.  2.2 1.5 1.4 2.3 2.4 1.8 1.8 2.1 2.4 2.3 1.9 2.3
 2.5 2.3 1.9 2.  2.3 1.8]
longp [1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 1.5 1.6 1.4 1.1 1.2 1.5 1.3 1.4
 1.7 1.5 1.7 1.5 1.  1.7 1.9 1.6 1.6 1.5 1.4 1.6 1.6 1.5 1.5 1.4 1.5 1.2
 1.3 1.5 1.3 1.5 1.3 1.3 1.3 1.6 1.9 1.4 1.6 1.4 1.5 1.4 4.7 4.5 4.9 4.
 4.6 4.5 4.7 3.3 4.6 3.9 3.5 4.2 4.  4.7 3.6 4.4 4.5 4.1 4.5 3.9 4.8 4.
 4.9 4.7 4.3 4.4 4.8 5.  4.5 3.5 3.8 3.7 3.9 5.1 4.5 4.5 4.7 4.4 4.1 4.
 4.4 4.6 4.  3.3 4.2 4.2 4.2 4.3 3.  4.1 6.  5.1 5.9 5.6 5.8 6.6 4.5 6.3
 5.8 6.1 5.1 5.3 5.5 5.  5.1 5.3 5.5 6.7 6.9 5.  5.7 4.9 6.7 4.9 5.7 6.
 4.8 4.9 5.6 5.8 6.1 6.4 5.6 5.1 5.6 6.1 5.6 5.5 4.8 5.4 5.6 5.1 5.1 5.9
 5.7 5.2 5.  5.2 5.4 5.1]
```



6) En supposant qu'il existe une relation de type $y = a*x+b$ liant la largeur (y) avec la longueur (x), trouver au moyen d'une technique d'ajustement de courbe, les valeurs optimales de a et b, de manière à ce que la droite passe au mieux par tous les points.

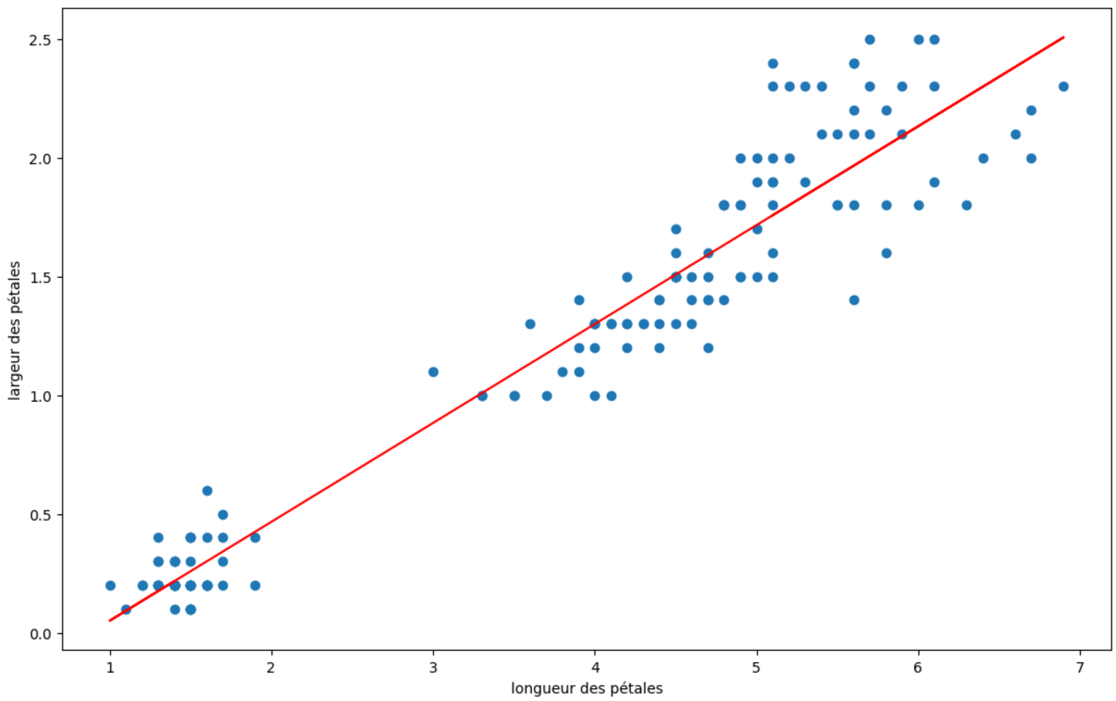
[9]:
```python
from scipy.optimize import curve_fit

# à compléter
```

```
on trouve a=  0.4164191327469227  et b=  -0.36651404713788666
```

7) Refaire le graphique en ajoutant la courbe déterminée ci-dessus sur le dessin.

[10]:
```python
# à compléter
```



8) Utiliser le résultat précédent pour donner une estimation de la largeur d'un pétale mesurant 2.5 cm de long

[11]:
```python
# à compléter
```

```
Estimation de la largeur d'un pétale mesurant 2.5 cm de long 0.6745337847294202
cm
```

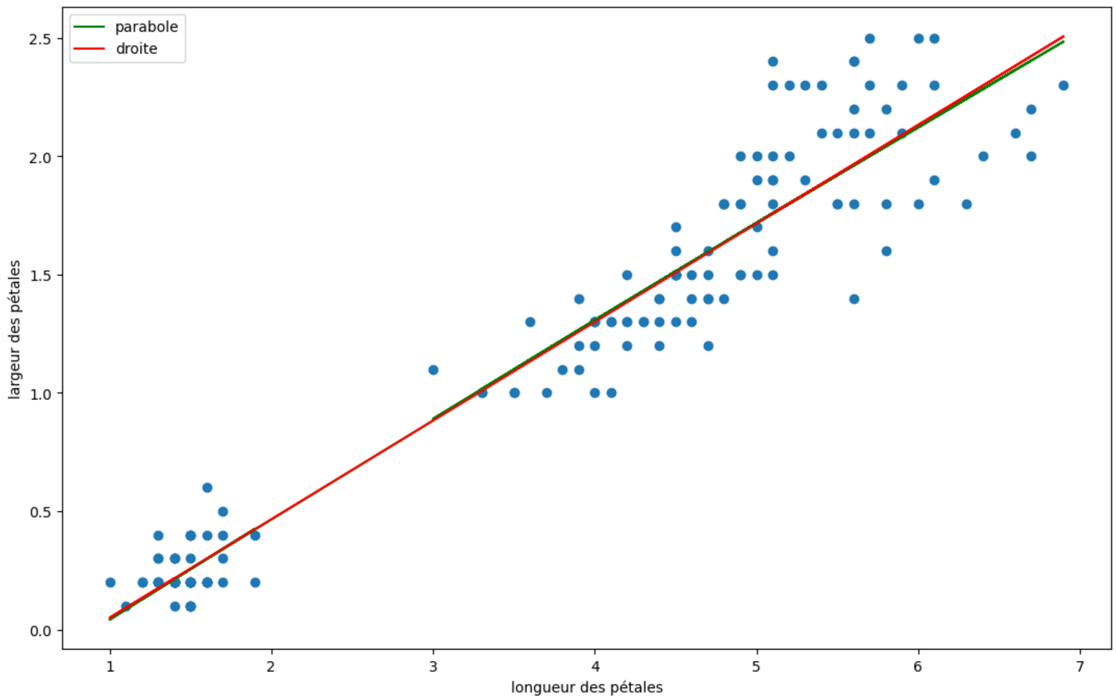
9) Quelqu'un pense qu'il faut plutôt prendre une courbe $y = cx^2 + dx + e$

Refaites l'optimisation et le dessin avec les deux courbes.

[12]:
```python
# à compléter
```

```
on trouve c=  -0.0026803025815989386  et d=  0.43528717556266244  et e=
-0.3912779199676333
```

[13]:
```python
# à compléter
```



10) Pour comparer calculer la somme des valeurs absolues des résidus c'est à dire $\sum |y_i - f(x_i)|$ pour chacune des fonctions.

[14]:
```python
# à compléter
```

```
23.588489197740785 23.73126621656111
```

3.1.2 Exercice 2:

1) Créer un tableau de taille 7x5, le remplir de valeurs aléatoires comprises entre 1 et 3 (compris) , calculer

- la somme de toutes les valeurs
- la somme de chaque colonne
- le nombre de 2 dans le tableau

[15]:
```python
import numpy as np
# à compléter
```

```
le tableau:
 [[1 3 3 2 3]
 [2 2 3 3 2]
 [2 3 2 1 3]
 [1 2 2 2 1]
 [3 1 3 2 2]
 [2 3 2 2 3]
 [3 1 1 2 2]]
Somme de toutes les valeurs: 75
Somme par colonnes: [14 15 16 14 16]
Nombre de 2: 16
```

2) Créer un tableau numpy de taille 10x10 et le remplir de valeurs aléatoires entières comprises entre 1 et 10 (compris).

- Remplacer toutes les valeurs inférieures ou égales à 2 par 0 et celle supérieures ou égales à 8 par 10.
- Mettre une colonne sur 3 à -1 (à partir de la colonne de gauche)
- Mettre un -2 dans les cases qui ont un indice pairs pour la ligne et pour la colonne

[16]:
```python
# à compléter
```

```
Le tableau :
 [[ 7  5 10  5  7  9  4  1 10  4]
 [ 3 10  3  4  1  6  3  3  8  8]
 [ 8 10  6  3  9  3  4  8  8  8]
 [10 10 10  8  7  7  5  6  5  8]
 [ 9  8  8  9  1  7  3 10  1  8]
 [ 6  4  3  3  8  8  4  2  7  2]
 [ 7  9  3  7  8  3  5  2  3  2]
 [ 3  6  7  8  2 10  2  5 10  8]
 [ 8  1  8  9 10  2  6  2 10  9]
 [10  8  9  4  1 10  9  4  7  1]]
```

[17]:
```python
# à compléter
```

```
Le tableau après première modification :
 [[ 7  5 10  5  7 10  4  0 10  4]
 [ 3 10  3  4  0  6  3  3 10 10]
 [10 10  6  3 10  3  4 10 10 10]
 [10 10 10 10  7  7  5  6  5 10]
 [10 10 10 10  0  7  3 10  0 10]
 [ 6  4  3  3 10 10  4  0  7  0]
 [ 7 10  3  7 10  3  5  0  3  0]
 [ 3  6  7 10  0 10  0  5 10 10]
 [10  0 10 10 10  0  6  0 10 10]
 [10 10 10  4  0 10 10  4  7  0]]
```

[18]:
```python
# à compléter
```

```
Le tableau après deuxième modification :
 [[-1  5 10 -1  7 10 -1  0 10 -1]
 [-1 10  3 -1  0  6 -1  3 10 -1]
 [-1 10  6 -1 10  3 -1 10 10 -1]
 [-1 10 10 -1  7  7 -1  6  5 -1]
 [-1 10 10 -1  0  7 -1 10  0 -1]
 [-1  4  3 -1 10 10 -1  0  7 -1]
 [-1 10  3 -1 10  3 -1  0  3 -1]
 [-1  6  7 -1  0 10 -1  5 10 -1]
 [-1  0 10 -1 10  0 -1  0 10 -1]
 [-1 10 10 -1  0 10 -1  4  7 -1]]
```

[19]:
```python
# à compléter
```

```
Le tableau après troisième modification :
 [[-2  5 -2 -1 -2 10 -2  0 -2 -1]
 [-1 10  3 -1  0  6 -1  3 10 -1]
 [-2 10 -2 -1 -2  3 -2 10 -2 -1]
 [-1 10 10 -1  7  7 -1  6  5 -1]
 [-2 10 -2 -1 -2  7 -2 10 -2 -1]
 [-1  4  3 -1 10 10 -1  0  7 -1]
 [-2 10 -2 -1 -2  3 -2  0 -2 -1]
 [-1  6  7 -1  0 10 -1  5 10 -1]
 [-2  0 -2 -1 -2  0 -2  0 -2 -1]
 [-1 10 10 -1  0 10 -1  4  7 -1]]
```

3.2 Partie II : sympy

3.2.1 Exercice 1 :

On considère les deux familles de fonctions suivantes :

$$f_k(x) = -0.05 * (x^3 + kx^2 + 4x + 1)$$

et

$$g_p(x) = \frac{\sin(x) - px}{x^3}$$

Où k et p sont des paramètres.

1) Définir à l'aide de sympy les deux expressions littérales correspondantes et afficher ces fonctions.

[20]:
```python
# à compléter
```

$-0.05kx^2 - 0.05x^3 - 0.2x - 0.05$

$$\frac{-px + \sin(x)}{x^3}$$

2) Calculer une primitive de f_k

[21]:
```python
# à compléter
```

```
Primitive de f :
```

$-0.0166666666666667kx^3 - 0.0125x^4 - 0.1x^2 - 0.05x$

3) Calculer la dérivée de f_k, puis déterminer les abscisses des extremums de la fonction

[22]:
```python
# à compléter
```

```
Dérivée de f :
```

$-0.1kx - 0.15x^2 - 0.2$

```
Abscisses des extremums de f :
```

$-0.333333333333333k - 1.15470053837925\sqrt{0.0833333333333333k^2 - 1}$

$-0.333333333333333k + 1.15470053837925\sqrt{0.0833333333333333k^2 - 1}$

4) Calculer la limite de $g_p(x)$ quand x tend vers 0 :

$\lim_{x\to 0} \frac{\sin(x)-px}{x^3}$.

Etudier le cas particulier où p vaut 1

[23]:
```python
# à compléter
```

```
Si p ne vaut pas 1  oo*sign(1 - p)
Si p vaut  1  -1/6
```

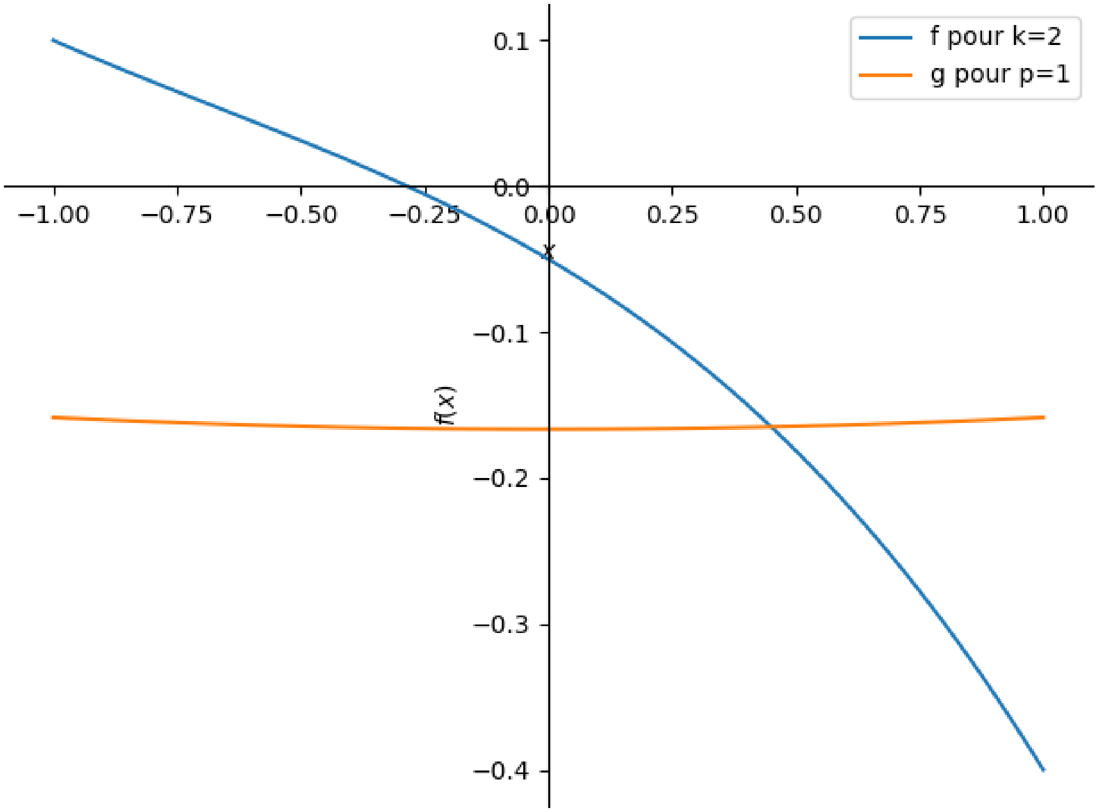
5) On étudie le cas où $p = 1$ et $k = 2$. Définir les fonctions $(f = f_2)$ et $(g = g_1)$ correspondantes

Tracer avec `sympy`, pour $x \in [-1, 1]$, le graphe des deux fonctions, sur le même graphique et avec légende comme ci-dessous.

[24]:
```python
# à compléter
```

$-0.05x^3 - 0.1x^2 - 0.2x - 0.05$

$$\frac{-x + \sin(x)}{x^3}$$



6) Déterminer avec sympy l'abscisse x_0 du point d'intersection de $f(x)$ et de l'axe des x.

[25]:
```
# à compléter
```

```
Abscisse de l'intersection : -0.284774761564910
```

7) On admettra que l'abscisse x_1 du point d'intersection des deux courbes (f et g) vaut 0.4505520891087474.

Déterminer l'aire comprise entre les deux courbes pour x variant entre x_0 et x_1.

[26]:
```
# à compléter
```

```
Aire obtenue : 0.0690292886928283
```

3.2.2 Exercice 2 :

A l'aide de la méthode de votre choix, résoudre le système suivant:

$$\begin{cases} 3x + 4y - z = 4 \\ x + 2y + 3z = 6 \\ 2x - y - z = -5 \end{cases}$$

[27]:
```
# à compléter
```

```
x=  -1 , y =  2 , z=  1
```

3.2.3 Exercice 3 :

Soit m un paramètre et M la matrice suivante :

$$M = \begin{pmatrix} 1 & 1 & -1 & 2 \\ 1 & 1 & m & 1 \\ -1 & 1 & 3 & 3 \\ 2 & 4 & 0 & m \end{pmatrix}$$

1. Définir la matrice M ci-dessus
2. Calculer le déterminant de M. On affichera le résultat puis le résultat factorisé.
3. Calculer les valeurs qui annulent le déterminant.
4. Quand m n'annule pas le déterminant calculer la matrice inverse de M.
5. Afficher la matrice inverse obtenue pour $m = 3$.

[28]:
```
# à compléter
```

```
Voici la matrice M:
```

$$\begin{bmatrix} 1 & 1 & -1 & 2 \\ 1 & 1 & m & 2 \\ -1 & 1 & 3 & 4 \\ 2 & 4 & 0 & m \end{bmatrix}$$

```
Déterminant de M:
```

$-2m^2 + 18m + 20$

```
Déterminant factorisé :
```

$-2\left(m - 10\right)\left(m + 1\right)$

```
valeurs qui annulent le déterminant :
```

```
[-1, 10]
```

[29]:
```
# à compléter
```

```
Matrice inverse quand elle a un sens:
```

$$\begin{bmatrix} \frac{m^2-19m+24}{2m^2-18m-20} & \frac{2}{m+1} & \frac{8-m}{2m-20} & \frac{1}{m-10} \\ \frac{m^2+11m-12}{2m^2-18m-20} & -\frac{1}{m+1} & \frac{m-4}{2m-20} & -\frac{3}{m-10} \\ -\frac{1}{m+1} & \frac{1}{m+1} & 0 & 0 \\ -\frac{6}{2m-20} & 0 & -\frac{2}{2m-20} & \frac{2}{2m-20} \end{bmatrix}$$

[30]:
```
# à compléter
```

la matrice obtenue pour m=3 :

[30]:

$$\begin{bmatrix} \frac{3}{7} & \frac{1}{2} & -\frac{5}{14} & -\frac{1}{7} \\ -\frac{15}{28} & -\frac{1}{4} & \frac{1}{14} & \frac{3}{7} \\ -\frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{3}{7} & 0 & \frac{1}{7} & -\frac{1}{7} \end{bmatrix}$$