TPnote15avril24

April 13, 2024

1 Calcul Scientifique – L1

2 TP Noté – le 15/4/2024

Travail réalisé par : NOM / PRENOM (à remplir impérativement!)

2.0.1 Instructions

1. Ce TP doit être fait sur les navigateurs des machines de l'université (**pas d'utilisation de VSCODIUM**) et individuellement.
2. Les accès à internet sont interdits à l'execption de ecampus (votre journal d'accès internet durant l'épreuve sera conservé) :

- http://ecampus.unicaen.fr

Donc pas d'accès à google ou à toute forme de messagerie ou de forums

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 et à l'intérieur des exercices certaines questions peuvent être indépendantes.

3 Exercice 1: travail avec numpy

Résoudre les questions de cet exercice à l'aide de fonctions de la librairie `numpy` sans utiliser ni de boucles `for` ou `while`, ni d'instructions `if`.

1. Créer un tableau de taille 8x8, le remplir de valeurs aléatoires comprises entre 1 et 7 (inclus). On va ensuite travailler sur ce tableau mais vos instructions doivent pouvoir fonctionner même si on changeait la taille du tableau.

Etant donné ce tableau, calculer : - la moyenne de tous les éléments du tableau - la somme des valeurs par lignes - la somme de la dernière colonne - le nombre de valeurs inférieures ou égales à la moyenne des valeurs du tableau - la somme des valeurs plus grandes strictement à 3 et inférieures ou égales à 5.

Remplacer les valeurs d'une ligne sur 4 par 0 en une seule ligne d'instruction, sans utiliser de boucles.

[1]:
```
import numpy as np

# Créer un tableau 8x8 rempli de valeurs aléatoires entre 1 et 7
# A compléter ......

# Calculer la moyenne de tous les éléments du tableau
# A compléter ......

# Calculer la somme des valeurs par lignes
# A compléter ......

# Calculer la somme de la dernière colonne
# A compléter ......

# Calculer le nombre de valeurs inférieures ou égales à la moyenne
# A compléter ......

# Calculer la somme des valeurs plus grandes strictement à 3 et inférieures ou␣
 ↪égales à 5
# A compléter ......
```

```
Tableau initial :


array([[5, 6, 7, 4, 6, 3, 3, 7],
       [1, 3, 2, 1, 3, 7, 1, 3],
       [5, 4, 5, 4, 1, 1, 7, 7],
       [6, 4, 3, 3, 3, 3, 1, 5],
       [5, 5, 7, 7, 5, 2, 6, 4],
       [2, 3, 3, 4, 7, 5, 1, 6],
       [7, 5, 5, 3, 4, 3, 6, 2],
       [5, 6, 3, 5, 6, 1, 3, 2]])

Moyenne de tous les éléments du tableau : 4.09375

Somme des valeurs par lignes : [41 21 34 28 41 31 35 31]

Somme de la dernière colonne : 36

Nombre de valeurs inférieures ou égales à la moyenne : 35

Somme des valeurs plus grandes strictement à 3 et inférieures ou égales à 5 : 19

Le tableau modifié:

array([[0, 0, 0, 0, 0, 0, 0, 0],
       [1, 3, 2, 1, 3, 7, 1, 3],
```

```
[5, 4, 5, 4, 1, 1, 7, 7],
[0, 0, 0, 0, 0, 0, 0, 0],
[5, 5, 7, 7, 5, 2, 6, 4],
[2, 3, 3, 4, 7, 5, 1, 6],
[0, 0, 0, 0, 0, 0, 0, 0],
[5, 6, 3, 5, 6, 1, 3, 2]])
```

2. Écrire une fonction qui prend en entrée une matrice (tableau 2D) et renvoie un nouveau tableau contenant les éléments uniques dans la matrice, c'est-à-dire les éléments qui n'apparaissent qu'une seule fois dans la matrice.

[7]:
```
# A compléter ......
print("Les éléments uniques dans la matrice sont :", resultat)
```

```
Les éléments uniques dans la matrice sont : [0 5 6 9]
```

4 Exercice 2 : Etude de données

Le tableau suivant représente les ventes de différents produits dans un magasin sur une période de 60 jours consécutifs. Chaque ligne du tableau représente une journée. Les colonnes ont la signification suivante :

```
- colonne 0 : numéro de semaine dans le mois  (valeurs possibles : 1.0, 2.0, 3.0, 4.0, 5.0)
- colonne 1 : numéro du jour de la semaine (1 pour lundi, 2 pour mardi, ..., 7 pour dimanche)
- colonne 2 : nombre de ventes du produit A
- colonne 3 : nombre de ventes du produit B
- colonne 4 : nombre de ventes du produit C
```

[2]:
```
# Le tableau représentant les ventes de la société sur 60 jours

ventes = [[ 1.   ,  3.   ,  61.543, 175.586, 302.448],
       [ 1.   ,  4.   ,  38.058,  56.037, 130.58 ],
       [ 1.   ,  5.   ,  21.826,  25.125,  82.461],
       [ 2.   ,  1.   ,  41.542, 113.294, 162.284],
       [ 2.   ,  2.   ,  37.679,  56.618, 116.22 ],
       [ 2.   ,  3.   ,  30.792,  50.704, 125.868],
       [ 2.   ,  4.   ,  43.304,  66.371, 153.368],
       [ 2.   ,  5.   ,  38.584,  85.961, 124.413],
       [ 3.   ,  1.   ,  33.973, 148.274, 162.044],
       [ 3.   ,  2.   ,  36.399,  43.306, 168.723],
       [ 3.   ,  3.   ,  45.706, 111.036, 124.678],
       [ 3.   ,  4.   ,  43.851,  66.277, 133.44 ],
       [ 3.   ,  5.   ,  43.339, 136.434, 128.405],
       [ 4.   ,  1.   ,  46.241, 120.865, 196.296],
       [ 4.   ,  2.   ,  56.519, 136.709, 143.644],
       [ 4.   ,  3.   ,  56.167,  78.101, 112.724],
       [ 4.   ,  3.   ,  51.66 ,  92.272, 164.948],
       [ 4.   ,  5.   ,  47.717,  71.474, 113.935],
```

```
       [  5.   ,   1.   ,  59.135, 157.681, 187.564],
       [  1.   ,   2.   ,  90.476,  80.509, 127.575],
       [  1.   ,   3.   ,  42.904,  43.962, 142.383],
       [  1.   ,   4.   ,  47.331,  72.444, 116.529],
       [  1.   ,   5.   ,  32.077, 127.358, 137.739],
       [  2.   ,   1.   ,  58.721, 139.034, 211.646],
       [  2.   ,   2.   ,  36.017,  75.813, 119.205],
       [  2.   ,   3.   ,  35.576,  79.997, 123.253],
       [  2.   ,   4.   ,  54.401,  75.613, 105.584],
       [  2.   ,   5.   ,  37.656,  59.907, 144.549],
       [  3.   ,   1.   ,  57.81 , 236.248, 196.732],
       [  3.   ,   2.   ,  43.359,  89.382, 156.916],
       [  3.   ,   3.   ,  45.555, 148.718, 104.186],
       [  3.   ,   4.   ,  45.55 , 120.548, 157.505],
       [  4.   ,   2.   ,  67.884, 267.342, 281.227],
       [  4.   ,   3.   ,  70.376, 154.242, 121.417],
       [  4.   ,   4.   ,  71.068, 100.544, 136.033],
       [  4.   ,   5.   ,  64.137, 109.062,  80.648],
       [  5.   ,   1.   , 118.178, 260.632, 152.134],
       [  5.   ,   2.   ,  51.199, 124.66 , 157.5  ],
       [  5.   ,   3.   ,  47.002,  99.892, 159.462],
       [  1.   ,   5.   , 109.888, 131.165, 175.777],
       [  2.   ,   1.   ,  77.388, 154.863, 182.936],
       [  2.   ,   2.   ,  46.295,  96.87 , 124.837],
       [  2.   ,   3.   ,  53.366,  69.15 , 111.987],
       [  2.   ,   4.   ,  47.399,  77.61 , 109.715],
       [  2.   ,   5.   ,  48.081,  72.826, 109.157],
       [  3.   ,   1.   ,  59.042, 130.098, 168.254],
       [  3.   ,   2.   ,  44.809,  99.072, 115.365],
       [  3.   ,   3.   ,  39.025, 110.74 ,  94.47 ],
       [  3.   ,   4.   ,  39.6  , 240.922, 122.085],
       [  3.   ,   5.   ,  57.467,  88.462, 109.132],
       [  4.   ,   1.   ,  41.418, 135.189, 165.999],
       [  4.   ,   2.   ,  34.193, 115.536, 118.911],
       [  4.   ,   3.   ,  32.653,  81.576,  74.372],
       [  4.   ,   4.   ,  51.985,  51.93 ,  98.107],
       [  4.   ,   5.   ,  36.748,  71.353, 105.408],
       [  5.   ,   1.   ,  59.131,  92.639, 165.079],
       [  5.   ,   2.   ,  54.224, 115.746, 116.442],
       [  5.   ,   3.   ,  58.378, 142.382, 102.687],
       [  5.   ,   4.   ,  76.763,  96.478, 131.709],
       [  5.   ,   5.   , 107.568, 121.152, 103.18 ]]
```

1) Après avoir converti le tableau en 'array' numpy, afficher les dimensions du tableau et calculer (en une ligne) la somme des ventes sur l'ensemble des jours.

Ce calcul doit se faire en une ligne de code, sans boucle.

[3]:
```
import numpy as np

# A compléter ......
```

```
Dimensions du tableau :  (60, 5)
somme ventes sur les 60 jours : 18052.399 €
```

2) Donner le nombre de jours où le montant des ventes dépasse les 350€.

[4]:
```
# A compléter ......
```

```
Nombre de jours où le montant des ventes dépasse 350€ : 11 journées
```

3) On veut faire des statistiques de vente selon les jours de la semaine.

Ecrire la fonction venteJour(i) qui renvoie le tableau des ventes du jour i. En déduire le tableau lundi qui correspond à ventejour(1)

[5]:
```
# A compléter ......
```

```
ventes du lundi  [[  2.      1.     41.542 113.294 162.284]
 [  3.      1.     33.973 148.274 162.044]
 [  4.      1.     46.241 120.865 196.296]
 [  5.      1.     59.135 157.681 187.564]
 [  2.      1.     58.721 139.034 211.646]
 [  3.      1.     57.81  236.248 196.732]
 [  5.      1.    118.178 260.632 152.134]
 [  2.      1.     77.388 154.863 182.936]
 [  3.      1.     59.042 130.098 168.254]
 [  4.      1.     41.418 135.189 165.999]
 [  5.      1.     59.131  92.639 165.079]]
ventes du mardi  [[  2.      2.     37.679  56.618 116.22 ]
 [  3.      2.     36.399  43.306 168.723]
 [  4.      2.     56.519 136.709 143.644]
 [  1.      2.     90.476  80.509 127.575]
 [  2.      2.     36.017  75.813 119.205]
 [  3.      2.     43.359  89.382 156.916]
 [  4.      2.     67.884 267.342 281.227]
 [  5.      2.     51.199 124.66  157.5  ]
 [  2.      2.     46.295  96.87  124.837]
 [  3.      2.     44.809  99.072 115.365]
 [  4.      2.     34.193 115.536 118.911]
 [  5.      2.     54.224 115.746 116.442]]
ventes du mercredi  [[  1.      3.     61.543 175.586 302.448]
 [  2.      3.     30.792  50.704 125.868]
 [  3.      3.     45.706 111.036 124.678]
 [  4.      3.     56.167  78.101 112.724]
 [  4.      3.     51.66   92.272 164.948]
 [  1.      3.     42.904  43.962 142.383]
```

```
 [  2.      3.     35.576  79.997 123.253]
 [  3.      3.     45.555 148.718 104.186]
 [  4.      3.     70.376 154.242 121.417]
 [  5.      3.     47.002  99.892 159.462]
 [  2.      3.     53.366  69.15  111.987]
 [  3.      3.     39.025 110.74   94.47 ]
 [  4.      3.     32.653  81.576  74.372]
 [  5.      3.     58.378 142.382 102.687]]
ventes du jeudi  [[  1.      4.     38.058  56.037 130.58 ]
 [  2.      4.     43.304  66.371 153.368]
 [  3.      4.     43.851  66.277 133.44 ]
 [  1.      4.     47.331  72.444 116.529]
 [  2.      4.     54.401  75.613 105.584]
 [  3.      4.     45.55  120.548 157.505]
 [  4.      4.     71.068 100.544 136.033]
 [  2.      4.     47.399  77.61  109.715]
 [  3.      4.     39.6   240.922 122.085]
 [  4.      4.     51.985  51.93   98.107]
 [  5.      4.     76.763  96.478 131.709]]
ventes du vendredi  [[  1.      5.     21.826  25.125  82.461]
 [  2.      5.     38.584  85.961 124.413]
 [  3.      5.     43.339 136.434 128.405]
 [  4.      5.     47.717  71.474 113.935]
 [  1.      5.     32.077 127.358 137.739]
 [  2.      5.     37.656  59.907 144.549]
 [  4.      5.     64.137 109.062  80.648]
 [  1.      5.    109.888 131.165 175.777]
 [  2.      5.     48.081  72.826 109.157]
 [  3.      5.     57.467  88.462 109.132]
 [  4.      5.     36.748  71.353 105.408]
 [  5.      5.    107.568 121.152 103.18 ]]
```

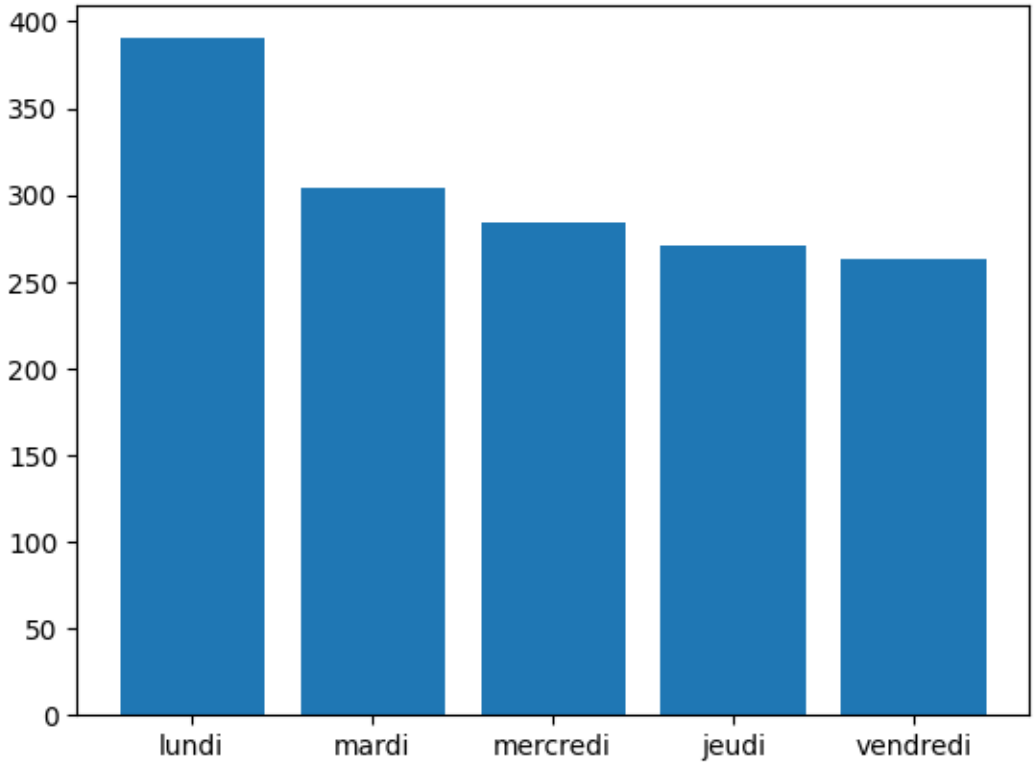
4) Ecrire la fonction moyenne(tab) qui étant donné un tableau du style de vente ou lundi renvoie la moyenne des ventes par jour pour ce tableau: il faut d'abord faire la somme par ligne puis faire la moyenne de ces valeurs

[6]:
```
# A compléter ......
```

```
pour les lundis 390.21490909090903
```

5) Faire un graphique pour comparer les moyennes des 5 jours lundi, mardi, mercredi , jeudi, vendredi comme ci-dessous

[7]:
```
import matplotlib.pyplot as plt
%matplotlib inline
fig,ax=plt.subplots()

# A compléter ......
```



6) On veut étudier la nature des ventes de catégorie A, B, C selon les jours. Construire (chacun en une ligne) les 3 tableaux donnant les montants des ventes pour les 60 jours respectivement pour les trois catégories A, B et C.

On pourra récupérer les réponses pour continuer l'exercice en cas d'échec à la question.

[8]:
```
# Construire les tableaux des ventes pour les 60 jours respectivement pour les␣
↪catégories A, B et C
# A compléter ......

print("Dépenses de catégorie A :\n",ventes_A)
```

```
Dépenses de catégorie A :
 [ 61.543  38.058  21.826  41.542  37.679  30.792  43.304  38.584  33.973
  36.399  45.706  43.851  43.339  46.241  56.519  56.167  51.66   47.717
  59.135  90.476  42.904  47.331  32.077  58.721  36.017  35.576  54.401
  37.656  57.81   43.359  45.555  45.55   67.884  70.376  71.068  64.137
 118.178  51.199  47.002 109.888  77.388  46.295  53.366  47.399  48.081
  59.042  44.809  39.025  39.6    57.467  41.418  34.193  32.653  51.985
  36.748  59.131  54.224  58.378  76.763 107.568]
```

[9]:
```
# Construire les tableaux des ventes pour les 60 jours respectivement pour les␣
 ↪catégories A, B et C
# A compléter ......

print("Dépenses de catégorie B :\n",ventes_B)
```

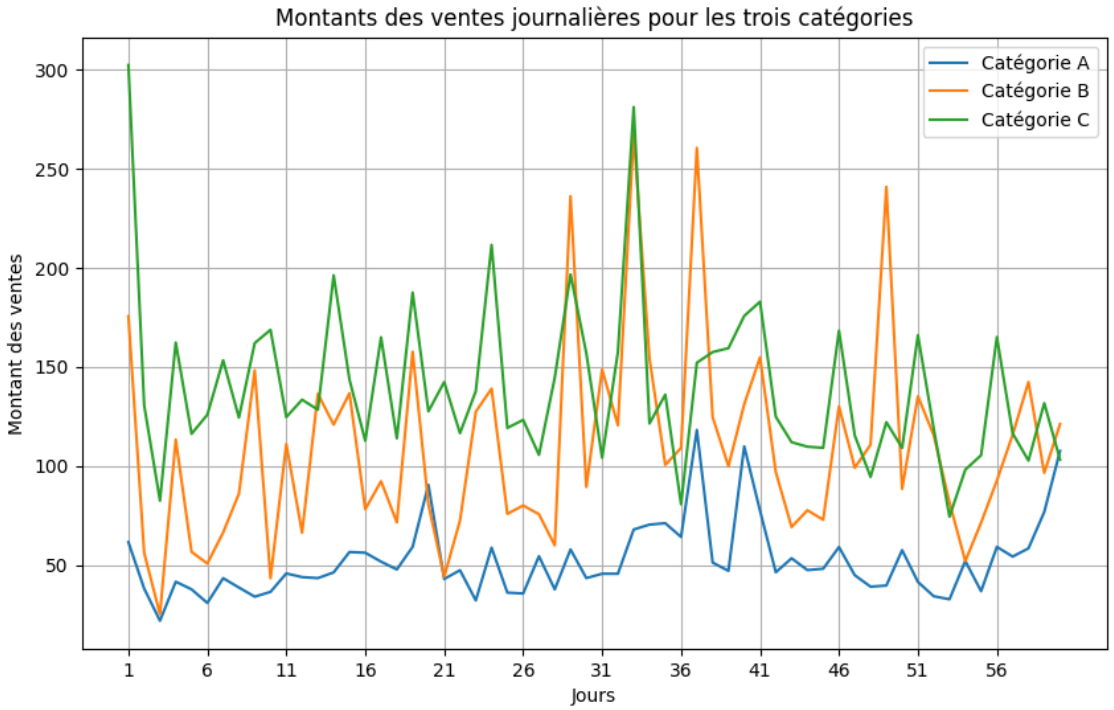
```
Dépenses de catégorie B :
 [175.586  56.037  25.125 113.294  56.618  50.704  66.371  85.961 148.274
  43.306 111.036  66.277 136.434 120.865 136.709  78.101  92.272  71.474
 157.681  80.509  43.962  72.444 127.358 139.034  75.813  79.997  75.613
  59.907 236.248  89.382 148.718 120.548 267.342 154.242 100.544 109.062
 260.632 124.66   99.892 131.165 154.863  96.87   69.15   77.61   72.826
 130.098  99.072 110.74  240.922  88.462 135.189 115.536  81.576  51.93
  71.353  92.639 115.746 142.382  96.478 121.152]
```

[10]:
```
# Construire les tableaux des ventes pour les 60 jours respectivement pour les␣
 ↪catégories A, B et C

# A compléter ......
print("Dépenses de catégorie C :\n",ventes_C)
```

```
Dépenses de catégorie C :
 [302.448 130.58   82.461 162.284 116.22  125.868 153.368 124.413 162.044
 168.723 124.678 133.44  128.405 196.296 143.644 112.724 164.948 113.935
 187.564 127.575 142.383 116.529 137.739 211.646 119.205 123.253 105.584
 144.549 196.732 156.916 104.186 157.505 281.227 121.417 136.033  80.648
 152.134 157.5   159.462 175.777 182.936 124.837 111.987 109.715 109.157
 168.254 115.365  94.47  122.085 109.132 165.999 118.911  74.372  98.107
 105.408 165.079 116.442 102.687 131.709 103.18 ]
```

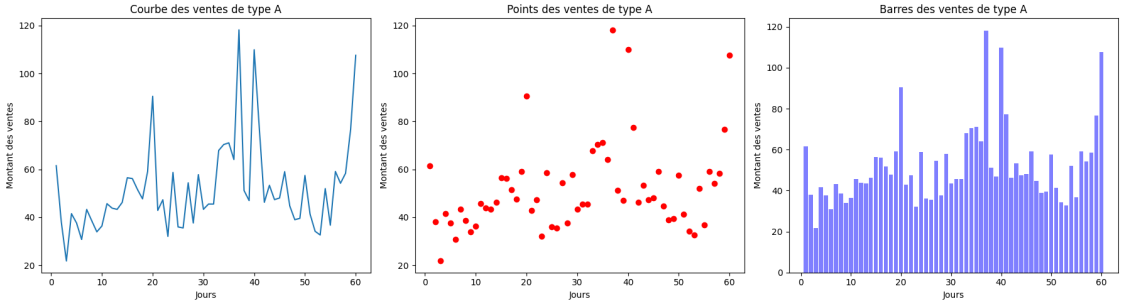
7) Tracer sur un même graphique les montants de ventes journalières pour les trois catégories (A, B, C). Votre shéma devra ressembler au shéma fourni.

[11]:
```
import matplotlib.pyplot as plt
%matplotlib inline

# A compléter ......
```



8) On s'intéresse de plus près aux ventes de type A. Faites un shéma comme donné ci-dessous avec les données des ventes de type A sous forme de courbe, de points et de barres.

[12]:
```
import matplotlib.pyplot as plt
%matplotlib inline


# A compléter ......
```



9) Nous nous intéressons uniquement à la courbe des ventes de type C au cours des 60 jours.

Le graphique précédent laisse supposer que les ventes suivent une loi de type

$$vente_C(t) = a * sin(b * t + c) + d$$

où a, b, c, d sont des paramètres.

Nous voulons vérifier cette hypothèse en réalisant un ajustement de courbe au moyen de scipy.

Trouver grace à cet ajustement de courbes les paramètres a, b, c, d optimaux.

[13]:
```
# A compléter ......
```

```
Paramètres optimaux :
a = -17.540387697967578
b = 1.1381152193292408
c = -9.125986663912741
d = 139.6269378039421
```

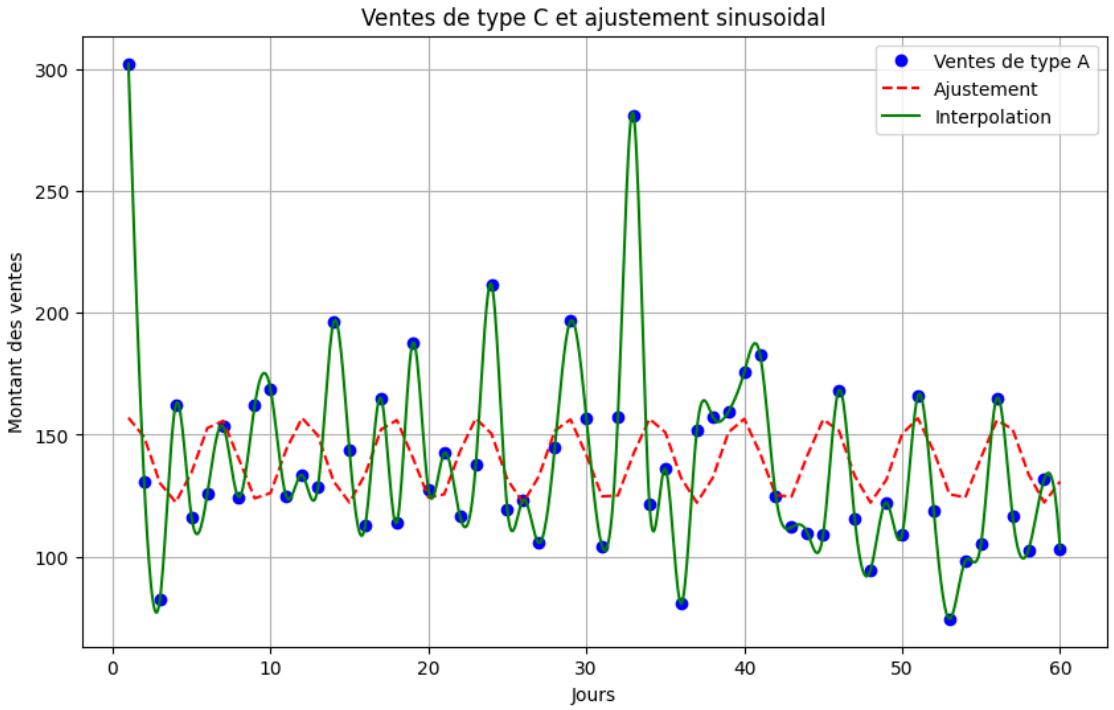
10) Calculer une mesure permettant d'évaluer l'écart entre les données et la courbe ajustée (résidus).

[14]:
```
# A compléter ......
```

```
Somme des valeurs absolues des écarts :  1780.8576893606078
```

11) Effectuer une interpolation des données de ventes de type A sur une période de 60 jours, par la fonction interp1d de la bibliothèque SciPy. Tracer sur le même graphique (voir figure ci-dessous) : la courbe des ventes de type A, celle de la fonction d'ajustement ainsi que celle de l'interpolation.

[15]:
```
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

## A compléter ......
```



5 Exercice 3 : calcul symbolique

On utilisera `sympy` pour cet exercice.

1) On considère les deux fonctions

$$f(x) = ax^2 + 2x/5 - 10$$

et

$$g(x) = \frac{\cos(x) - x^2}{2} + kx$$

où a et k sont des paramètres réels.

Définir à l'aide de sympy les deux expressions littérales correspondantes.

[16]:
```
import sympy as sy
# A compléter ......
```

`fa=`

$$ax^2 + \frac{2x}{5} - 10$$

`gk=`

$$kx - \frac{x^2}{2} + \frac{\cos(x)}{2}$$

2) Calculer les limites en $+\infty$ de f_a et g_k

[17]:
```
# A compléter ......
```

pour g -oo
pour f oo*sign(a)

3) Calculer la dérivée de fa et trouver les abscisses qui annulent cette dérivée

[18]:
```
# A compléter ......
```

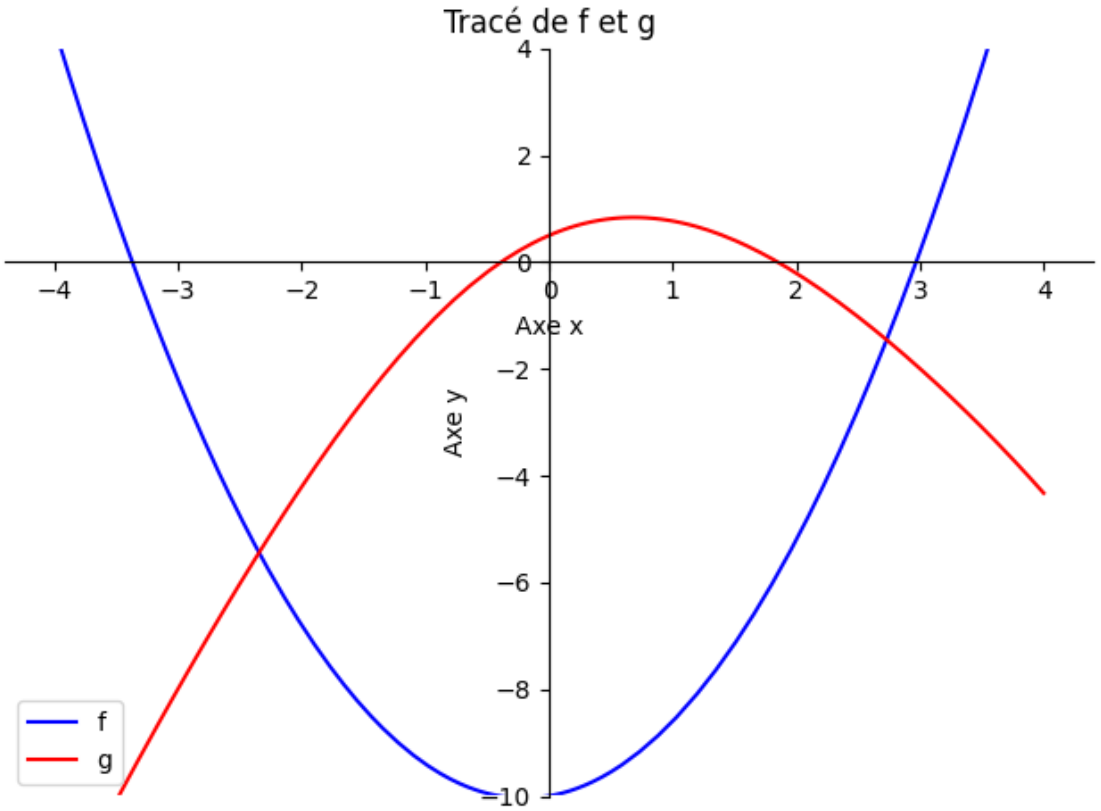
$2ax + \frac{2}{5}$

[18]:
```
[-1/(5*a)]
```

4) On s'intéresse à $f = f_1$ et $g = g_1$

Tracer avec `sympy`, pour $x \in [-4, 4]$, et $y \in [-10, 4]$, le graphe des deux fonctions, sur le même graphique et avec légende comme ci-dessous.

[19]:
```
# A compléter ......
```



5) Déterminer avec sympy le nombre de points d'intersection de $f(x)$ et de l'axe des x. Affichez les différents abscisses des points d'intersection.

Vous utiliserez pour cela la résolution de l'équation $f(x) = 0$.

[20]:
```
# Résolution de l'équation

# A compléter ......
```

```
Nombre d'intersections avec l'axe des abscisses : 2
Points d'intersection avec l'axe des x :
x_0 = 2.96859590355097
x_1 = -3.36859590355097
```

6) Solve ne permet pas de déterminer les intersections de f et g. En définissant une fonction Python correspondant à $f(x) - g(x)$, utilisez la fonction fsolve du module scipy.optimize pour déterminer les valeurs de x aux points d'intersection des deux courbes.

Utilisez fsolve(fct, xdep) où fct est la fonction à résoudre et xdep est un tableau de points de départ des itérations. Nous prendrons la liste [-2.5, 2.5] comme points de départ.

[21]:
```
import scipy as sc

# A compléter ......
```

```
Positions de l'intersection: x0= -2.3443604604889097  et x1= 2.7300523219549246
```

7) Déterminer l'aire (en valeur absolue) comprise entre les deux courbes pour x variant entre x0 et x1.

[22]:
```
# A compléter ......
```

```
Aire entre les deux courbes: 35.2728725763035
```

6 Exercice 4 : calcul matriciel avec numpy et sympy

6.1 A . Système avec numpy

On veut résoudre le système suivant :

$$\begin{cases} x + 2y - z & = 2 \\ -x - 3y + z & = 0 \\ x + 3y + 2z & = 1 \end{cases}$$

1. Si on écrit ce systeme sous la forme $A \cdot X = B$, définir les matrices A et B
2. Calculer le déterminant de la matrice A
3. Calculer la solution X de deux manières différentes (soit avec un calcul de matrice soit avec la résolution directe) On donnera les résultats approchés avec un chiffre après la virgule.

[23]:
```
# A compléter ......
```

```
Matrice A:
Matrice B:
Déterminant de A: -3.0

Solution 1 avec arrondi : x= 6.3  y= -2.0  z= 0.3

Solution 2 avec arrondi : x= 6.3  y= -2.0  z= 0.3
```

6.2 B. Matrices et sympy

Soit a un paramètre et M la matrice suivante :

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

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 a n'annule pas le déterminant calculer la matrice inverse de M.
5. Afficher la matrice inverse obtenue pour $a = 2$.

[24]:
```
# A compléter ......
```

```
Voici la matrice M:
```

$$\begin{bmatrix} 1 & a & a^2 & a^3 \\ a & a^2 & a^3 & 1 \\ a^2 & a^3 & 1 & a \\ a^3 & 1 & a & a^2 \end{bmatrix}$$

```
Déterminant de M:
```

$a^{12} - 3a^8 + 3a^4 - 1$

```
Déterminant factorisé :
```

$(a-1)^3 (a+1)^3 (a^2+1)^3$

```
valeurs réelles qui annulent le déterminant :

[-1, 1]
```

[25]:
```
# A compléter ......
```

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

$$\begin{bmatrix} \frac{1}{1-a^4} & 0 & 0 & -\frac{a}{1-a^4} \\ 0 & 0 & \frac{a}{a^4-1} & -\frac{1}{a^4-1} \\ 0 & \frac{a}{a^4-1} & -\frac{1}{a^4-1} & 0 \\ -\frac{a}{1-a^4} & \frac{1}{1-a^4} & 0 & 0 \end{bmatrix}$$

[26]:
```
# A compléter ......
```

```
la matrice obtenue pour a=2 :
```

[26]:
$$\begin{bmatrix} -\frac{1}{15} & 0 & 0 & \frac{2}{15} \\ 0 & 0 & \frac{2}{15} & -\frac{1}{15} \\ 0 & \frac{2}{15} & -\frac{1}{15} & 0 \\ \frac{2}{15} & -\frac{1}{15} & 0 & 0 \end{bmatrix}$$

7 Exercice 5 : logique

On veut travailler sur la formule F définie par:

$F = (p \vee \neg(q \vee r)) \rightarrow \neg\Big((\neg r \rightarrow q) \wedge \neg(p \vee q)\Big)$

Définir F avec sympy.

[27]:
```
import sympy as sp

# A compléter ......
```

$(p \vee \neg\,(q \vee r)) \Rightarrow \neg\,((\neg r \Rightarrow q) \wedge \neg\,(p \vee q))$

Que vaut F si on a p=True, q=False et r=False?

[28]:
```
# A compléter ......
```

[28]:
True

Déterminer une formule équivalente à F écrite sous forme Conjonctive.

[29]:
```
# A compléter ......
```

```
Forme conjonctive : (p | q | r | ~q) & (p | q | r | ~r) & (p | q | ~p | ~q) & (p
| q | ~p | ~r)
```

Faire une table de vérité pour la formule F comme ci-dessous.

[30]:
```
# A compléter ......
```

```
pour   (False, False, False) on obtient  True
pour   (False, False, True) on obtient  True
pour   (False, True, False) on obtient  True
pour   (False, True, True) on obtient  True
pour   (True, False, False) on obtient  True
pour   (True, False, True) on obtient  True
```

```
pour   (True, True, False) on obtient  True
pour   (True, True, True) on obtient  True
```