

PROGRAMMES, ENTRÉES ET SORTIES

1 Programmes

Le programme donne le PGCD de a par b .

2 Fonctions

- 1.
2. On obtient un message d'erreur (si la longueur de L1 est plus grande que celle de L2).
- 3.

```
def fois(L1, L2):
    n = len(L1)
    R = []
    for i in range(n):
        x = L1[i] * L2[i]
        R.append(x)
    return R
```

4.

```
def somme(L):
    n = len(L)
    S = 0
    for i in range(n):
        S += L[i]
    return S
```

5.

```
def moyenne(L):
    return somme(L)/len(L)
```

6.

```
def contient(L, x):
    n = len(L)
    for i in range(n):
        if x == L[i]:
            return True
    return False
```

7. Les fonctions plus, fois, somme et moyenne ont un coût linéaire car nécessitent un parcours complet de la liste.

3 Lecture de fichier

- 1.
2. On rajoute la ligne :

```
print(somme(y))
```

en fin du programme. Dans la console, on obtient :

```
>>> (executing file "TPl.py")
566.3960999999998
```

3.

```
n = len(x)
sx = somme(x)
sy = somme(y)
sxx = somme(fois(x,x))
sxy = somme(fois(x,y))
mx = moyenne(x)
my = moyenne(y)

a = (n * sxy - sx * sy) / (n*sxx -sx**2)
b = my - a * mx
```

4.

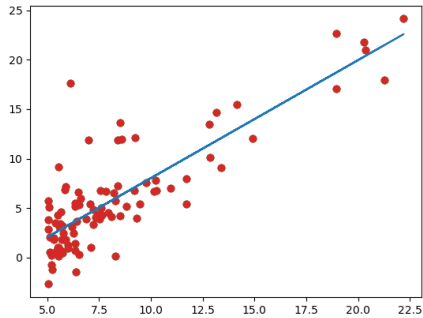
```
import matplotlib.pyplot as plt

plt.scatter(x,y)

def model(x,a,b):
    y = []
    for i in range(len(x)):
        y.append(a * x[i] + b)
    return y

yModel = model(x,a,b)
plt.plot(x,yModel)
plt.show()
```

On obtient le graphique suivant :



5.

```
plt.close()
f=open('notes.txt','r')
data=f.readlines()
X = []
for chaine in data:
    X.append(float(chaine.strip()))
f.close()

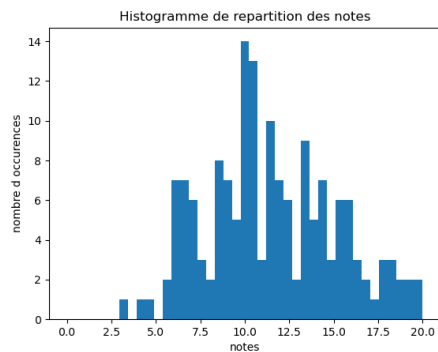
print("La moyenne de l'examinateur est de : ", moyenne(X),".")
```

On trouve une moyenne de 11,53.

BONUS : on peut tracer l'histogramme de la répartition des notes :

```
plt.figure()
plt.hist(X,range = (0, 20), bins = 41)
plt.title('Histogramme de repartition des notes')
plt.xlabel('notes')
plt.ylabel('nombre d occurrences')
plt.show()
```

On obtient l'histogramme suivant :



4 Écriture de fichier

1.

```
f = open('text.txt','w')
f.write('Hello world !')
f.close()
```

2. (a)

```
from math import *

def correction(gamma,x):
    note = (x**gamma)*(20**(1-gamma))
    arrondi = (ceil(note*10))/10
    return arrondi
```

(b)

```
f=open('notes.txt','r')
data=f.readlines()
X = []
for chaine in data:
    X.append(float(chaine.strip()))
f.close()

gamma = 0.8
f = open('notes2.txt','w')
n = len(X)
for i in range(n):
    x_cor = correction(gamma,X[i])
    f.write(str(x_cor) + '\n')
f.close()
```

Fin.