Python 5
Régression linéaire

L'objectif de cette fiche est d'effectuer des ajustements affines par la méthode des moindres carrés dans deux situations concrètes.
Les bibliothèques **numpy** et **matplotlib.pyplot** seront utiles.

I Notes de 11 étudiants d'ECG

On considère un groupe de 11 étudiants de CPGE ECG dont voici la moyenne en mathématiques en deuxième année de CPGE ECG (les x_i) ainsi que la moyenne aux écrits de mathématiques de la BCE (les y_i).

x_i	4,3	6,4	12,1	13,2	8,8	15	7,3	9,6	10,1	11,9	5,5
y_i	9	12,5	16,2	17,5	14,1	19,7	13	12,8	14,1	15,6	10,9

1. Recopier le code suivant :

```python
import matplotlib.pyplot as plt

x=[4.3,6.4,12.1,13.2,8.8,15,7.3,9.6,10.1,11.9,5.5]
y=[9,12.5,16.2,17.5,14.1,19.7,13,12.8,14.1,15.6,10.9]
plt.plot(x,y,"b+")
plt.xlabel("moyennes en CPGE")
plt.ylabel("moyennes aux écrits BCE")
plt.show()
```

2. Calculer les moyennes et écart-types des séries statistiques **x** et **y**.
3. Faire figurer le point moyen sur le graphique.
4. Déterminer la covariance ainsi que le coefficient de corrélation linéaire des séries statistiques **x** et **y**. Que peut-on en dire ?
5. Déterminer l'équation réduite de la droite d'ajustement affine obtenue par la méthode des moindres carrés, exprimant **y** en fonction de **x**.
6. La représenter sur le graphique.

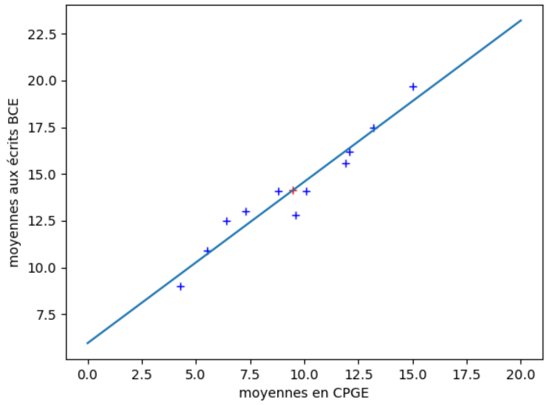
☞ Rappel...
Les commandes **np.mean** et **np.std** permettant d'obtenir moyenne et écart-type d'une liste ou d'un tableau.

Le programme complet...

```python
import matplotlib.pyplot as plt
import numpy as np

x=[4.3,6.4,12.1,13.2,8.8,15,7.3,9.6,10.1,11.9,5.5]
y=[9,12.5,16.2,17.5,14.1,19.7,13,12.8,14.1,15.6,10.9]
plt.plot(x,y,"b+")
plt.xlabel("moyennes en CPGE")
plt.ylabel("moyennes aux écrits BCE")

mx=np.mean(x)
my=np.mean(y)
sx=np.std(x)
sy=np.std(y)
cov=np.mean([(x[i]-mx)*(y[i]-my) for i in range(0,len(x))])
rho=cov/(sx*sy)

a=cov/(sx**2)
b=my-a*mx

plt.plot(mx,my,"r+")
plt.plot([0,20],[b,a*20+b])
plt.show()
```

Petite remarque
En transformant x et y en tableau (avec **x=np.array(x)**), on peut calculer la covariance en effectuant **np.mean((x-mx)*(y-my))**. En effet, **x-mx** permet de soustraire **mx** à chaque valeur de **x**, puis on effectue le produit terme à terme avec le tableau **y-my**...

La droite d'ajustement affine par la méthode des moindres carrés est la droite d'équation $y = ax + b$, avec $a \simeq 0,86$ et $b \simeq 5,96$.
Le coefficient de corrélation linéaire vaut environ $0,97$: l'ajustement est très bon.
Avec le graphique :



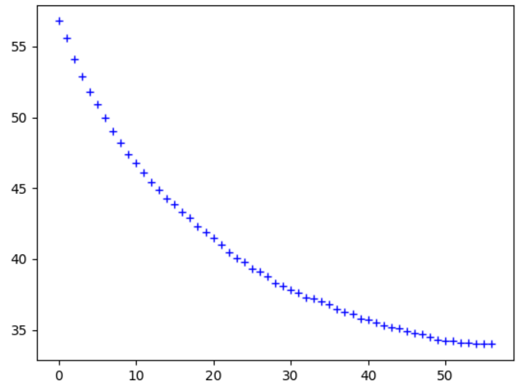
II Évolution de température

Lors d'une chaude journée de mai 2022, on a mesuré la température d'un café en fonction du temps. Le tableau de données est à retrouver sur Teams.

1. Représenter le graphique donnant la température en fonction du temps.
2. Comment la température du café semble-t-elle dépendre en fonction du temps ?
3. Effectuer un ajustement affine par la méthode des moindres carrés afin d'obtenir une expression de la température du café en fonction du temps et représenter la courbe obtenue sur le graphie initial de données.

✘ Attention !
L'ajustement affine en question ne s'effectue pas entre la température et le temps...

La courbe :



Le lien entre la température (notée y) et le temps (noté x) n'est pas du tout affine.
En revanche, on va chercher une relation de la forme :

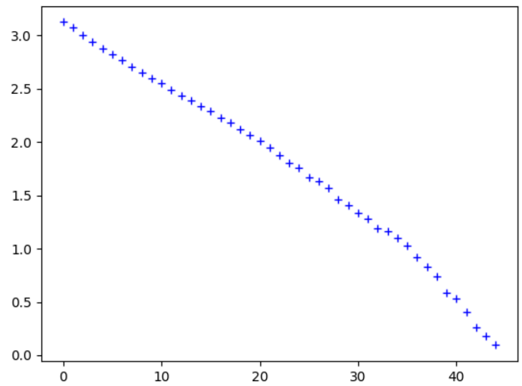
$$y = \beta e^{\alpha x} + 34$$

avec $\alpha < 0$ et $\beta > 0$ Autrement dit :

$$\ln(y - 34) = \alpha x + \ln(\beta)$$

Représentons donc $\ln(y - 34)$ en fonction de x :

Pourquoi ?
- 34 semble être la limite en $+\infty$...
- allure d'une exponentielle décroissante, d'où $\alpha < 0$
- , température supérieure à 34, d'où $\beta > 0$



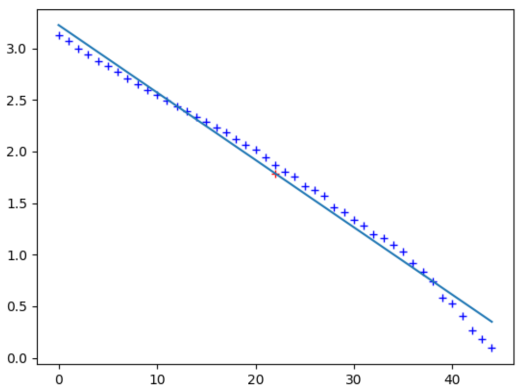
Le programme pour obtenir l'ajustement affine :

```python
import matplotlib.pyplot as plt
import numpy as np

y=np.array([56.8,55.6,54.1,52.9,51.8,50.9,50,49,48.2,47.4,46.8,46.1,45.4,44.9,44.3,43.9,43.3,42.9,42.3,41.
x=range(len(y))
plt.plot(x,y,"b+")
plt.xlabel("temps")
plt.ylabel("température du café")
plt.show()

Y=np.log(np.array([e for e in y if e>=35])-34)
X=range(len(Y))
plt.plot(X,Y,"b+")

mX=np.mean(X)
mY=np.mean(Y)
sX=np.std(X)
sY=np.std(Y)
cov=np.mean([(X[i]-mX)*(Y[i]-mY) for i in range(0,len(X))])
rho=cov/(sX*sY)

alpha=cov/(sX**2)
beta=mY-alpha*mX

plt.plot(mX,mY,"r+")
plt.plot([0,max(X)],[beta,alpha*max(X)+beta])
plt.show()

plt.plot(x,y,"b+")
Labs=np.linspace(min(x),max(x),1000)
plt.plot(Labs,np.exp(alpha*Labs+beta)+34)
plt.xlabel("temps")
plt.ylabel("température du café")
plt.show()
```

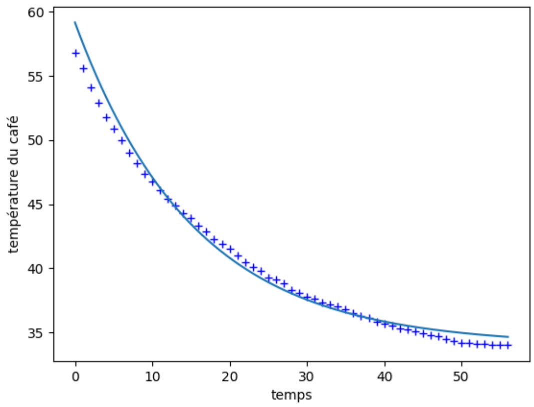
★Subtile...★

On exclut les valeurs trop proches de 34 pour éviter que $\ln(y-34)$ parte vers $-\infty$... L'ajustement final sera sans doute moins bon autour des grandes valeurs du temps...

La droite d'ajustement affine par la méthode des moindres carrés est la droite d'équation $Y = ax+b$, avec $a \simeq -0,065$ et $b \simeq 3,23$.
Le coefficient de corrélation linéaire vaut environ -0.99 : l'ajustement est très bon.
La droite des moindres carrés :



Et le modèle collé au nuage de point initial


température du café
temps
60
55
50
45
40
35
0
10
20
30
40
50