# TP3correction

February 9, 2020

## 1 TP3 - Traitement de données avec numpy et matplotlib

Pour commencer, on inclut (avec import) dans le bloc de code les librairies `numpy` et `matplotlib` :

```
In [1]: %matplotlib inline
        import matplotlib as mpl
        import matplotlib.pyplot as plt
        import numpy as np
```

### 1.1 Première partie, tracé de courbes avec numpy

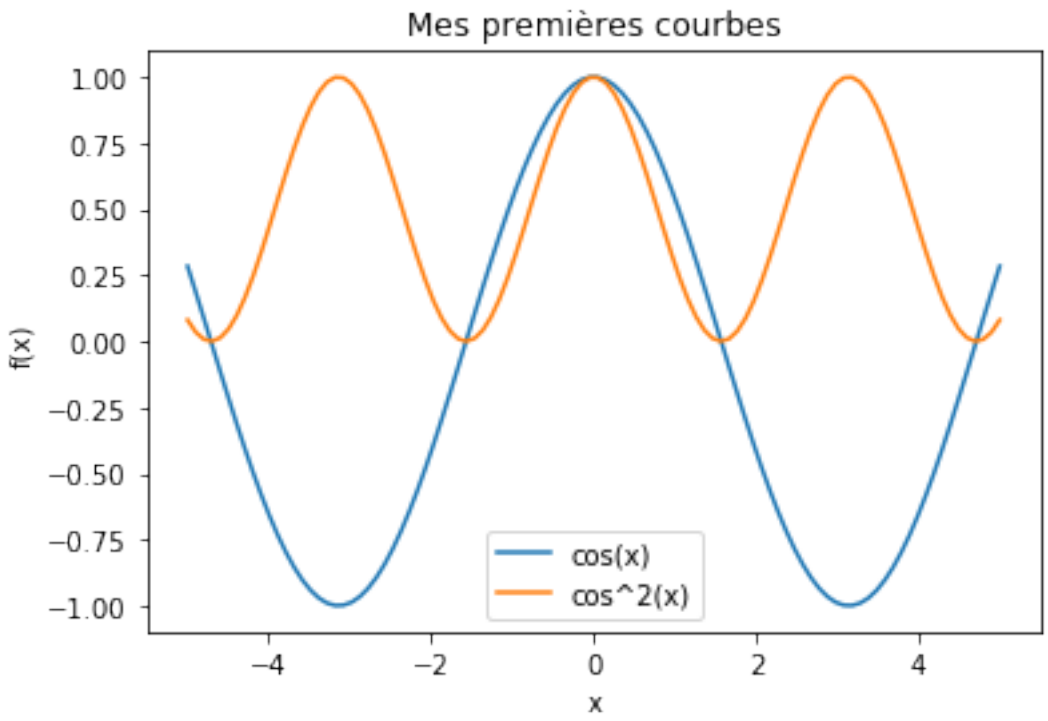
Tracer, sur un même graphique, le graphe des fonctions $\cos(x)$ et $\cos^2(x)$, pour des valeurs de $x$ comprises entre -5 et 5, par pas de 0.1

Ajouter les légendes et les propriétés de style de manière à ce que le résultat soit conforme à ce qui vous est donné ci-dessous. Il est obligatoire d'utiliser numpy et matplotlib.

```
In [2]: x = np.linspace(-5,5,100)
        f1 = np.cos(x)
        f2 = np.cos(x)**2

        fig, ax = plt.subplots()
        ax.plot(x,f1,label="cos(x)")
        ax.plot(x,f2,label="cos^2(x)")
        ax.set_xlabel("x")
        ax.set_ylabel("f(x)")
        ax.legend()
        ax.set_title("Mes premières courbes")

Out[2]: Text(0.5, 1.0, 'Mes premières courbes')
```



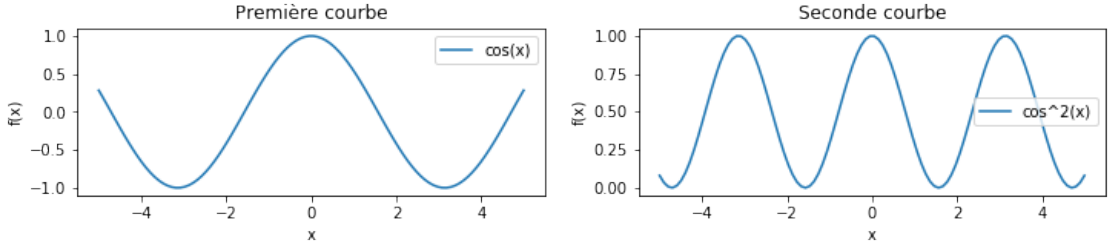

Tracer les mêmes courbes, mais cette fois-ci chacune sur un shéma sur deux zones côte à côte comme sur le shéma ci-dessous:

```
In [3]: fig, ax = plt.subplots(ncols=2, nrows=1,figsize=(12, 2))
        ax[0].plot(x,f1,label="cos(x)")
        ax[1].plot(x,f2,label="cos^2(x)")
        ax[0].set_xlabel("x")
        ax[0].set_ylabel("f(x)")

        ax[1].set_xlabel("x")
        ax[1].set_ylabel("f(x)")

        ax[0].legend()
        ax[1].legend()
        ax[0].set_title("Première courbe")
        ax[1].set_title("Seconde courbe")
```

```
Out[3]: Text(0.5, 1.0, 'Seconde courbe')
```



Sauvegarder le résultat précédent dans le fichier `figure.pdf`

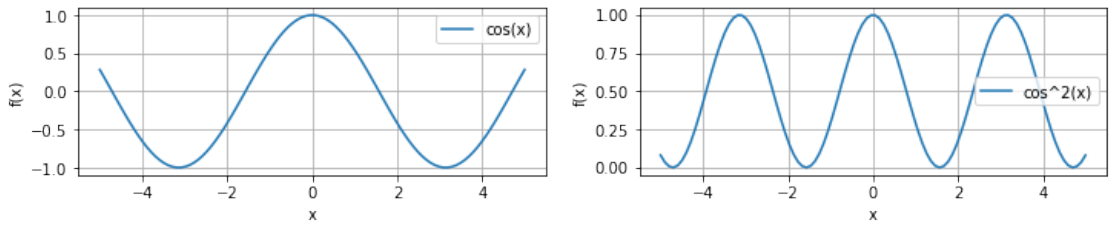
```
In [4]: fig.savefig("figure.pdf")
```

Reprenez les shémas ci-dessus, mais en appliquant une grille sur la figure.

```
In [5]: fig, ax = plt.subplots(ncols=2, nrows=1,figsize=(12, 2))
        ax[0].plot(x,f1,label="cos(x)")
        ax[1].plot(x,f2,label="cos^2(x)")
        ax[0].set_xlabel("x")
        ax[0].set_ylabel("f(x)")

        ax[1].set_xlabel("x")
        ax[1].set_ylabel("f(x)")

        for a in ax:
            a.grid()
            a.legend()
```



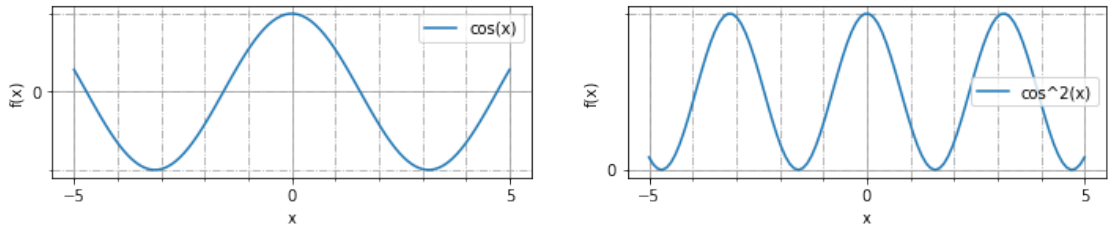

Puis en rajoutant avec des 'tick' principaux à 5 et des ticks secondaires à 1. On utilisera des commandes du style a.xaxis.set_major_locator(mpl.ticker.MultipleLocator(5))
On pourra utiliser une boucle pour ne pas écrire deux fois la même chose!

```
In [6]: fig, ax = plt.subplots(ncols=2, nrows=1,figsize=(12, 2))
        ax[0].plot(x,f1,label="cos(x)")
        ax[1].plot(x,f2,label="cos^2(x)")
        ax[0].set_xlabel("x")
        ax[0].set_ylabel("f(x)")

        ax[1].set_xlabel("x")
        ax[1].set_ylabel("f(x)")

        for a in ax:
            a.grid()
            a.legend()
            a.xaxis.set_major_locator(mpl.ticker.MultipleLocator(5))
            a.yaxis.set_major_locator(mpl.ticker.MultipleLocator(5))
```

```
a.xaxis.set_minor_locator(mpl.ticker.MultipleLocator(1))
a.yaxis.set_minor_locator(mpl.ticker.MultipleLocator(1))
a.grid(b=True, which='minor', color='0.65',linestyle='-.')
a.grid(b=True, which='major', color='0.65',linestyle='-')
```




## 1.2 Seconde partie : Analyse de données

Le tableau qui suit contient le relevé des températures de 1947 à 2018 à St-Etienne Bouthéon. Il se compose de 12 lignes (les mois) et 72 colonnes (les années).

```
In [7]: temp = [[-3.6,1.2,-0.5,-1.6,0.8,-3.2,-4.7,-2.1,2.2,0.4,-4.7,-0.3,-1.1,-1.9,-0.2,1.6,-7
```

Créer, à partir du tableau python `temp` défini ci-dessus, une variable numpy de type `array`, nommée `t` contenant les valeurs du tableau, puis afficher les dimensions de `t` pour vérifier que vous avez bien 12 lignes et 72 colonnes.

```
In [8]: t = np.array(temp)
        print(t.shape)
```

```
(12, 72)
```

### 1.2.1 Extraction et affichage des statistiques simples

Remarque: pour cette partie, il n'y a aucune boucle à écrire.

Quelle est l'année pour laquelle la température de janvier a été la plus froide ? Quelle était cette température ?

```
In [9]: print('température la plus froide observée en janvier : ',np.min(t[0,:]))
        print('c\'était en : ',1947+np.argmin(t[0,:]))
```

```
température la plus froide observée en janvier :  -7.3
c'était en :  1985
```

Calculer la température moyenne sur toutes les années et tous les mois. Afficher la valeur approchée avec 1 chiffre significatif.

```
In [10]: print('température moyenne : ', np.round(np.mean(t),1))
```

```
température moyenne :  6.0
```

Calculer le tableau `ta` contenant les valeurs moyennes des températures de chaque année. Calculer le tableau `tm`contenant les temparatures moyenne de chaque mois, sur l'ensemble des années. Afficher ces deux tableaux (approximation à 1 chiffre significatif).

```
In [11]: ta = np.mean(t, axis=0)
         tm = np.mean(t, axis=1)
         print('temp annuelles : ', np.round(ta))
         print('temp mensuelles : ', np.round(tm))
```

```
temp annuelles :  [6. 6. 6. 6. 6. 5. 5. 5. 5. 4. 5. 6. 6. 6. 6. 4. 4. 6. 5. 6. 5. 5. 5. 5.
 4. 4. 4. 5. 5. 5. 6. 5. 6. 5. 5. 6. 6. 5. 5. 6. 6. 7. 6. 7. 6. 7. 6. 8.
 7. 6. 7. 6. 7. 7. 7. 8. 7. 7. 6. 7. 7. 7. 7. 6. 7. 7. 6. 8. 7. 7. 6. 7.]
temp mensuelles :  [-1. -0.  2.  4.  8. 11. 13. 13. 10.  7.  3.  0.]
```

Calculer la température minimale (`tmin`) maximale (`tmax`) et médiane (`tmed`) pour chaque mois de l'année.

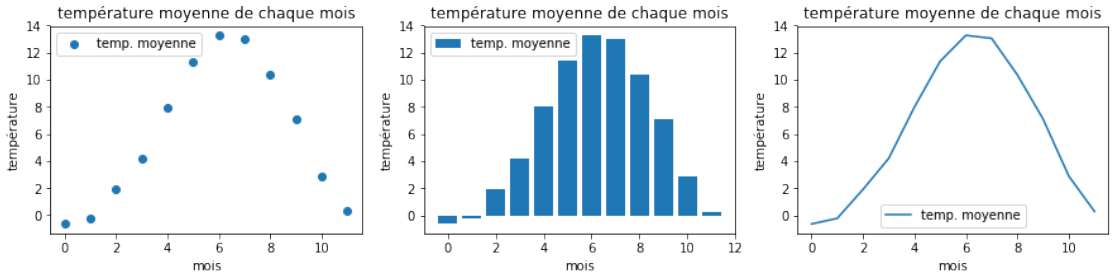
```
In [12]: tmin = np.min(t, axis=1)
         tmax = np.max(t, axis=1)
         tmed = np.median(t, axis=1)
         print("tmin", tmin)
         print("tmax", tmax)
         print("tmed", tmed)
```

```
tmin [ -7.3 -12.8  -3.2   0.2   4.9   7.8  10.3  10.7   5.1   3.1  -0.6  -5.2]
tmax [ 3.7  4.6  6.5  7.2 11.4 16.9 17.6 17.8 14.1 11.2  6.6  4.9]
tmed [-0.3   0.05  2.    4.55  7.9  11.4  13.25 13.05 10.2   7.05  2.55  0.4 ]
```

Affichage de la température moyenne des 12 mois de l'année. Faire une figure contenant 3 sous-figures représentant ces températures de 3 manières différentes, à l'image de la figure qui suit.

```
In [13]: f,ax = plt.subplots(1,3,figsize=(15,3))
         ax[0].scatter(np.arange(0,12),tm,label='temp. moyenne')
         ax[0].set(title='température moyenne de chaque mois',ylabel='température',xlabel='mois
         ax[0].legend(loc='best')
         ax[1].bar(np.arange(0,12),tm,label='temp. moyenne')
         ax[1].set(title='température moyenne de chaque mois',ylabel='température',xlabel='mois
         ax[1].legend(loc='best')
         ax[2].plot(np.arange(0,12),tm,label='temp. moyenne')
         ax[2].set(title='température moyenne de chaque mois',ylabel='température',xlabel='mois
         ax[2].legend(loc='best')
```

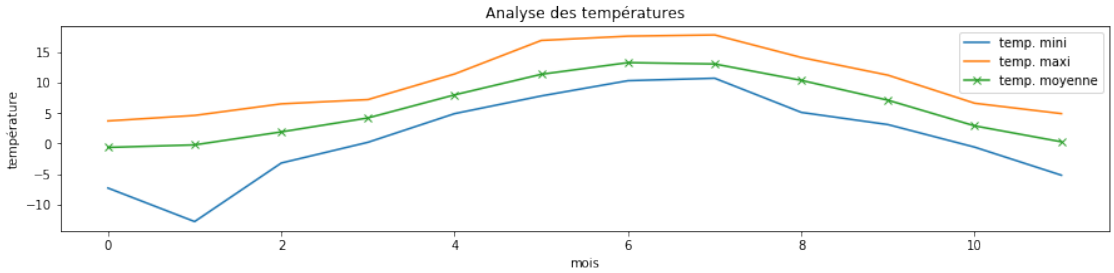
```
Out[13]: <matplotlib.legend.Legend at 0x7ff40bba7908>
```




Faire un graphique où figurent les températures moyennes, min et max comme sur le shéma ci-dessous.

```
In [14]: f,ax = plt.subplots(1,1,figsize=(15,3))
         ax.plot(np.arange(0,12),tmin,label='temp. mini')
         ax.plot(np.arange(0,12),tmax,label='temp. maxi')
         ax.plot(np.arange(0,12),tm,label='temp. moyenne',marker='x')
         ax.set(title='Analyse des températures',ylabel='température',xlabel='mois')
         ax.legend(loc='best')
```

```
Out[14]: <matplotlib.legend.Legend at 0x7ff40baa2630>
```




### 1.2.2 Calcul d'histogrammes

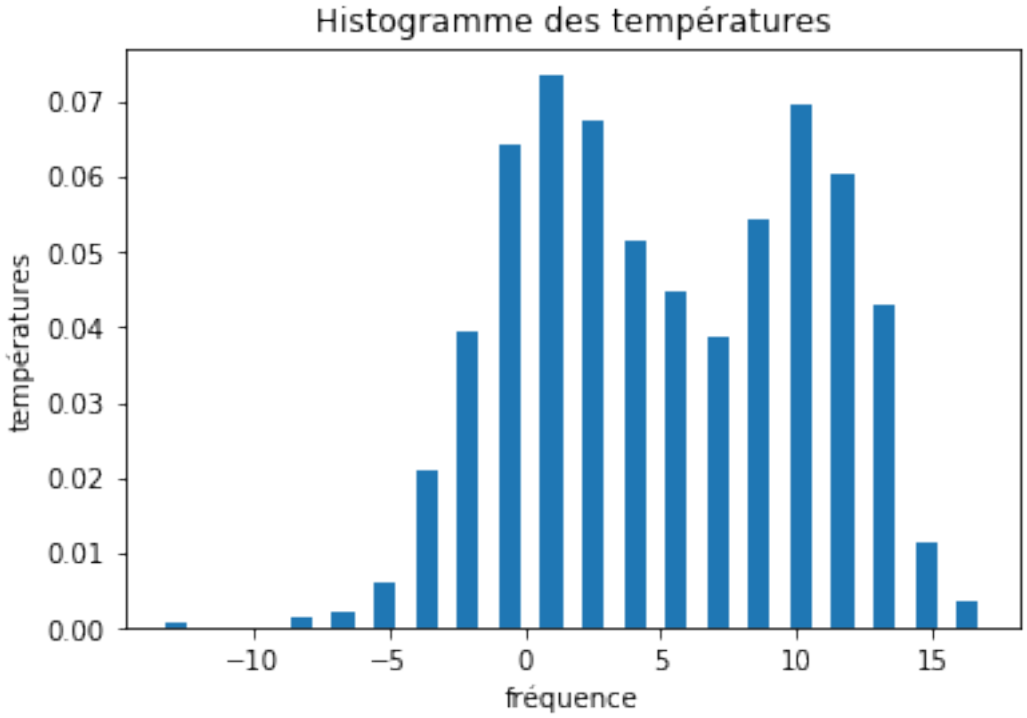
Calculer l'histogramme des température sur l'ensemble de la période considérée au moyen de la fonction `histogram`de `numpy`. Regarder dans la documentation quelles sont les paramètres de cette fonctions et ce qu'elle retourne.

```
In [15]: h = np.histogram(t, bins=20, density=True)
         print(h)
```

```
(array([0.00075648, 0.        , 0.        , 0.00151295, 0.00226943,
       0.0060518 , 0.02118131, 0.03933672, 0.06430041, 0.07337812,
       0.06732631, 0.05144033, 0.04463205, 0.03858025, 0.05446623,
       0.06959574, 0.06051803, 0.0431191 , 0.01134713, 0.00378238]), array([-12.8 , -11.27,  -9
        -0.56,   0.97,   2.5 ,   4.03,   5.56,   7.09,   8.62,  10.15,
        11.68,  13.21,  14.74,  16.27,  17.8 ]))
```

Afficher cet histogramme avec `matplotlib`.

```
In [16]: fig, ax = plt.subplots()
         ax.set(title='Histogramme des températures',ylabel='températures',xlabel='fréquence')
         plt.bar(h[1][0:-1],h[0])
         plt.show()
```




### 1.2.3 Mesures de dispersion des températures.

Nous souhaitons désormais mesurer la dispersion des températures, c'est-à-dire regarder comment les valeurs se répartissent autour de la moyenne. Les valeurs min et max ne donnent pas une bonne indication de la dispersion. La variance et l'écart type sont liés à la manière dont les points diffèrent de la moyenne. Plus la variance est grande, plus la dispersion est forte. La variance se calcule avec la formule :

$$variance = \frac{\sum_i (x_i - x_{moy})^2}{n}$$

L'écart-type est la racine carré de la variance :

$$ect = \sqrt{variance}$$

On commence par calculer la variance et l'écart type des températures de chaque mois de l'année. Il y aura donc 12 variances/écarts types. Dans un premier temps, on fera les calculs en utilisant les opérateurs de base de numpy (np.sum(),-,/,**2), sans faire de boucle.

Afficher les résultats obtenus.

```
In [17]: tvar = np.sum((t-tm.reshape(-1,1))**2,axis=1)/t.shape[1]
         tstd = np.sqrt(tvar)
         print('Ecart type des températures de chaque mois',tstd)
```

```
Ecart type des températures de chaque mois [2.33653055 2.71891487 1.74749324 1.4845806  1.5077
 1.54530289 1.38848814 1.59347568 1.91948263 1.75914717 2.02943188]
```

Utiliser ensuite les fonctions dédiées de numpy ( var et std) et vérifier que les résultats sont identiques.

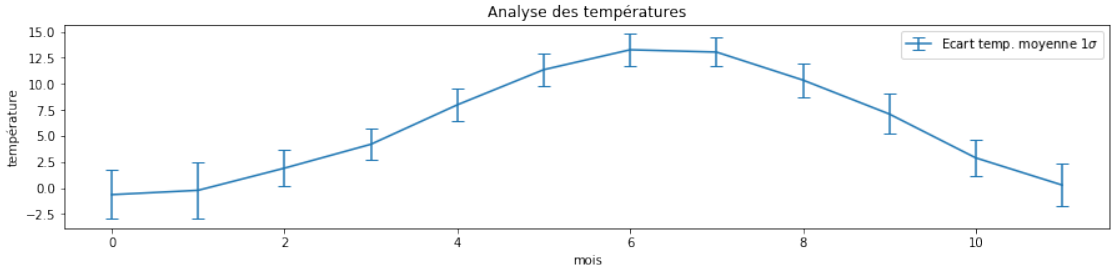
```
In [18]: tvar = np.var(t,axis=1)
         tstd = np.std(t,axis=1)
         print(tstd)
```

```
[2.33653055 2.71891487 1.74749324 1.4845806  1.50777819 1.59295323
 1.54530289 1.38848814 1.59347568 1.91948263 1.75914717 2.02943188]
```

Tracer un graphique contenant les températures mensuelles ainsi que des barres d'erreurs indiquant la dispersion à $[-\sigma,\sigma]$ autour des moyennes où $\sigma$ désigne l'écart-type.

```
In [19]: f,ax = plt.subplots(1,1,figsize=(15,3))
         ax.set(title='Analyse des températures',ylabel='température',xlabel='mois')
         ax.errorbar(np.arange(0,12),tm,yerr = tstd,label='Ecart temp. moyenne $1\sigma$',caps
         ax.legend(loc='best')
```

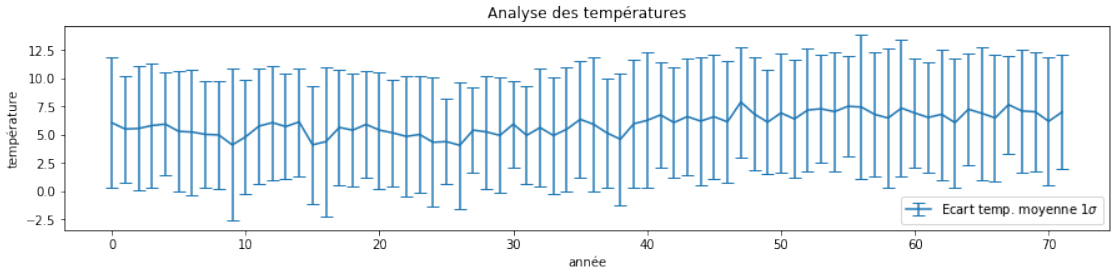
```
Out[19]: <matplotlib.legend.Legend at 0x7ff40b9f2eb8>
```




Faire le même graphique donnant la dispersion des températures pour chaque année.

```
In [20]: f,ax = plt.subplots(1,1,figsize=(15,3))
         ax.set(title='Analyse des températures',ylabel='température',xlabel='année')
         ax.errorbar(np.arange(0,len(ta)),ta,yerr = 1*np.std(t,axis=0),label='Ecart temp. moye
         ax.legend(loc='best')
```

```
Out[20]: <matplotlib.legend.Legend at 0x7ff40bb65a58>
```




### 1.2.4 Mesure de corrélation entre des séries de données

Dans cette section, nous allons regarder comment mesurer statistiquement la relation entre deux séries de nombres, en mesurant le coefficient de corrélation de Pearson entre les deux séries. Ce coefficient mesure la force de la relation linéaire qui lie les deux séries. Ce coefficient peut être positif, négatif ou nul. Son amplitude varie dans l'intervalle [-1, 1]. 0 indique que les séries ne sont pas corréllées, 1 (ou une valeur proche) qu'elles sont positivement corrélée, -1 qu'elles sont négativement correllées. Ce coefficient se mesure avec la formule :

$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}}$$

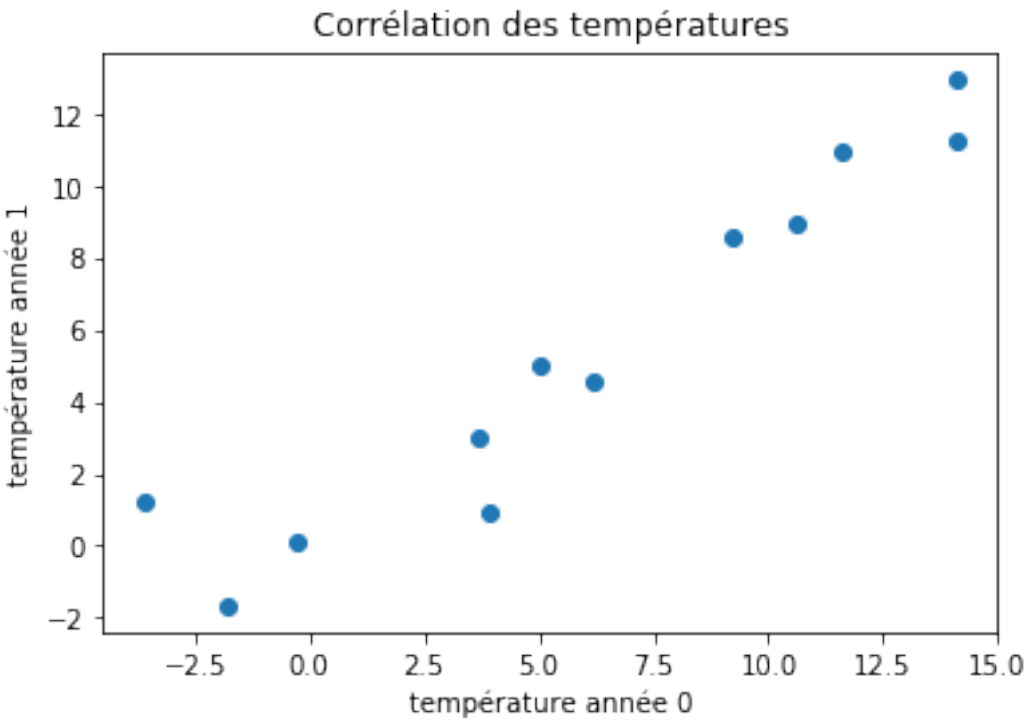
Nous allons chercher à mesure la corrélation entre les températures de la première et de la seconde année enregistrée, pour les 12 mois de l'année.

Réaliser un graphique qui donne la température de la seconde année en fonction de celle de la première, pour les 12 mois (on travaillera donc avec mes deux premières lignes du tableau) .

Calculer la corrélation de Pearson entre les séries `t[:,0]` et `t[:,1]` Remarque, cette fonction est également disponible dans la librairie `scipy.stats.stats`, fonction `pearsonr`.

```
In [21]: fig, ax = plt.subplots()
         ax.scatter(t[:,0],t[:,1])
         ax.set(title='Corrélation des températures',ylabel='température année 1',xlabel='tempe
         x=t[:,0]
         mx=np.mean(x)
         y=t[:,1]
         my=np.mean(y)
         c = np.sum((x-mx)*(y-my)) /np.sqrt((np.sum((x-np.mean(x))**2)*np.sum((y-np.mean(y))**2
         print('corrélation entre les deux série : ', c)
```

```
corrélation entre les deux série :  0.951800083464439
```


Corrélation des températures
température année 1
température année 0
12
10
8
6
4
2
0
−2
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0