Scipy

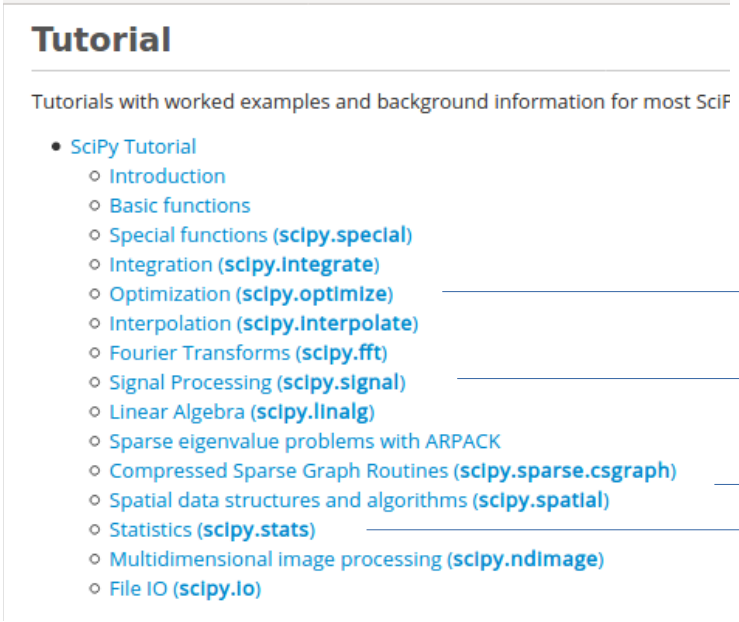
Cours M2 CCN

Modélisation des problèmes scientifiques

2019/2020

À quoi ça sert ?

Librairie python d'algorithmes pour résoudre des problèmes scientifiques (tests statistiques, traitement du signal, optimisation, etc.)



Capture d'écran du site de scipy

Optimisation

Traitement du signal

Graphes

Tests statistiques

En résumé, c'est beaucoup de maths (algèbre linéaire, résolution d'équations différentielles, intégrales, polynômes, transformée de fourier, etc.)

Dans ce cours, on se contentera des deux thèmes soulignés, qui sont plus abordables.

Plan

1- Optimisation

2- Tests statistiques

1- Optimisation

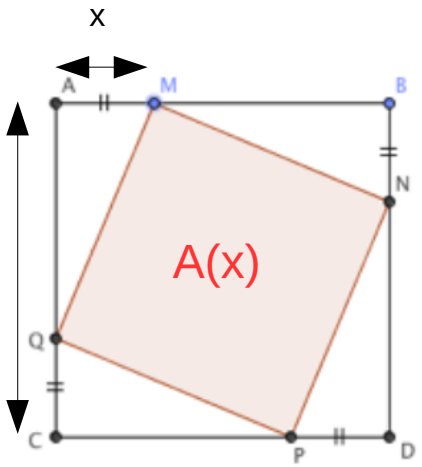
Qu'est-ce que l'optimisation ?

« L'optimisation est une branche des mathématiques cherchant à modéliser, à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à minimiser ou maximiser une fonction sur un ensemble. »

Ok, mais quel rapport avec les données ?

Exemple de problème d'optimisation:

La mesure du côté du carré ABCD est un réel positif noté a



Trouver la valeur xmin de x
telle que A(x) (l'aire de MNPQ) soit minimale pour a=5.

On appelle A(x) la fonction de coût
(parfois fonction objectif).

Pour $0<x<a$, $A(x) = x^2 + (a-x)^2$
On cherche alors xmin = argmin{A(x)}, $0<x<a$

```
Entrée [17]: from scipy.optimize import minimize

Entrée [18]: a = 5

             def A(x):
                 res = 0
                 if x <= a:
                     res = x**2+(x-a)**2
                 return res

Entrée [19]: minimize(A,x0=0) #x0 est la valeur de départ

Out[19]:       fun: 12.500000000000002
             hess_inv: array([[0.25]])
                  jac: array([1.1920929e-07])
              message: 'Optimization terminated successfully.'
                 nfev: 9
                  nit: 2
                 njev: 3
               status: 0
              success: True
                    x: array([2.50000002])
```

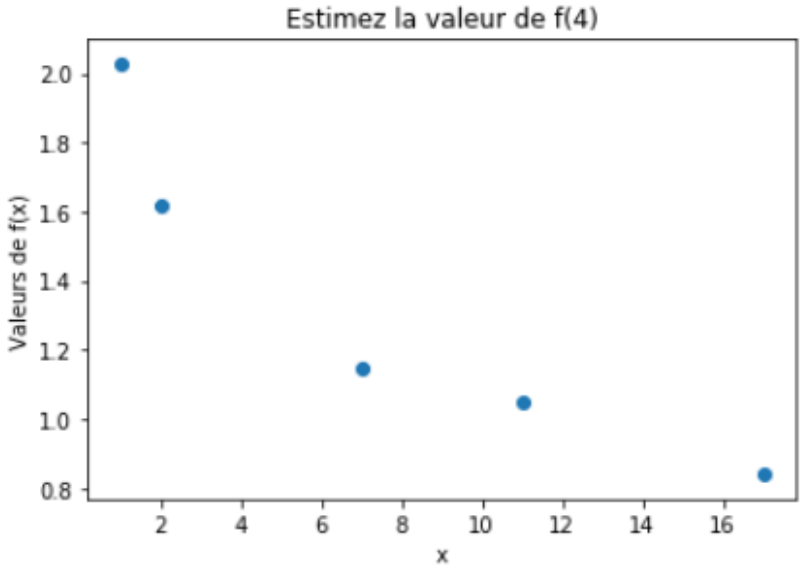
A(xmin)

xmin

Optimisation ;
Vers le machine learning

```
x = [1,2,7,11,17]
fx = [2.02483571, 1.61759158, 1.14796408, 1.05027089, 0.84315297] #f(x)

plt.scatter(x, fx)
plt.ylabel("Valeurs de f(x)")
plt.xlabel("x")
plt.title("Estimez la valeur de f(4)")
plt.show()
```



On sait que f est de la forme $f(x) = \frac{\beta}{x^{\alpha}}$, α et β à déterminer.

Dans votre jeu de données, vous disposez des valeurs de f(x) pour certaines valeurs de x (1,2,7,11,17).

Trouvez des valeurs plausibles pour α et β. Puis déterminez la valeur de f(4).

C'est de l'interpolation de valeurs, on cherche à trouver f(4), en prenant en compte que f est paramétrique, et grâce à un échantillon de valeurs de f

<u>Activité :</u>

Fonction de coût ;

$$F(\alpha,\beta)=\sum_{i\in 1,2,7,11,17}\left(f(i)-\frac{\beta}{i^{\alpha}}\right)^2$$

1) Trouvez alpha_min et bêta_min, les arguments qui minimisent F(alpha, bêta)

2) En déduire f

3) Prédire f(4)

Optimisation ;
Vers le machine learning

SciPy

Définition des fonctions

```python
def fab(t, ab):
    # input : t, ab est ici le couple de paramètres (alpha, beta)
    # output : f(t), avec alpha = ab[0] et beta = ab[1]
    # avec alpha = 1 et beta = 2 au point x = 1
    return ab[1]/(t**ab[0])

# par exemple, fab(1,(2,3)) retourne 3/1² = 3
# print(fab(1,(2,3)))

def F(ab):
    # la fonction à minimiser
    return sum([(fx[i] - fab(x[i], ab))**2 for i in range(len(x))])
```

Minimisation

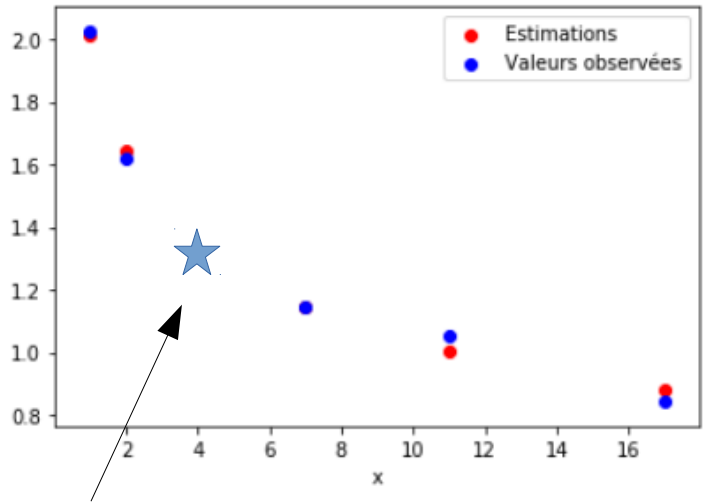
```python
alpha_min, beta_min = minimize(F, x0=(0,2.5))['x']
minimize(F, x0=(0,2.5))
```

```
      fun: 0.004839900763815356
 hess_inv: array([[0.05008846, 0.09250532],
       [0.09250532, 0.37715111]])
      jac: array([ 7.10249878e-07, -2.33470928e-07])
  message: 'Optimization terminated successfully.'
     nfev: 48
      nit: 9
     njev: 12
   status: 0
  success: True
        x: array([0.29103498, 2.01264793])
```

Résultats

```python
def f_estimee(t):
    return fab(t,(alpha_min, beta_min))

plt.scatter(x,
            [f_estimee(k) for k in x],
            c='red',
            label='Estimations')
plt.scatter(x,
            fx,
            c='blue',
            label='Valeurs observées')
plt.legend()
plt.xlabel('x')
plt.show()
```




```python
f_estimee(4)
```

```
1.344458175802727
```

2- Tests statistiques

Pour lever une ambiguïté ;

- Test statistiques ≠ tests logiciels (unitaires, d'intégration, etc.)

Mais il y a des similarités ; si les tests unitaires vérifient que les fonctions fonctionnent correctement, les tests statistiques s'intéressent aux propriétés des données, et cherchent à prouver des hypothèses à l'aide de méthodes probabilistes.

Pour comprendre à quoi sert un test statistique, on s'intéresse à un exemple ; le test du chi-deux (d'adéquation).

Attention, cette partie (4 prochaines slides) est assez théorique, elle existe pour montrer le fonctionnement complet d'un test statistique dans le détail. En pratique, une ligne de code suffira pour lancer un test.

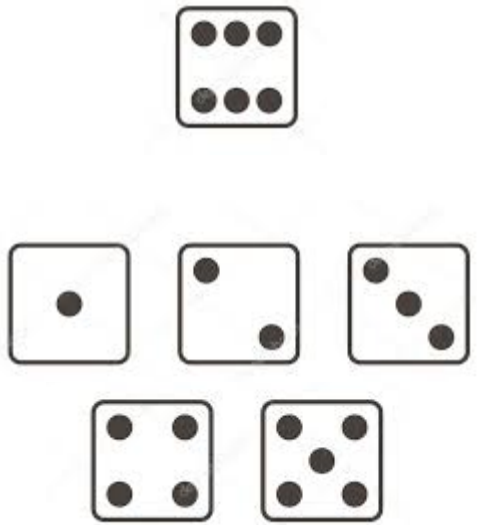
Test statistique du chi-2: Intuition

<u>Pari</u> :

Je lance un dé (à six faces).

Si j'obtiens un six, vous me payez deux cafés.

Si j'obtiens un autre chiffre, je vous paie un café.



1/6 → +2*

5/6 → -1*

Sur le papier, je ne pars pas gagnant.

À chaque fois que je lance le dé pour un nouveau pari, je perds en moyenne un demi café.

L'espérance du pari est définie par :

E(pari) = (1/6)*(+2 cafés) + (5/6)*(-1 café)
= 2/6-5/6 café
= -1/2 café

Test statistique du chi-2: Intuition (2)

Mais, en jouant 10 fois à ce jeu, je tombe quatre fois sur le chiffre 6, gagnant ainsi 2 cafés !

Vous vous demandez si je n'ai pas triché, et pipé le dé pour qu'il tombe plus souvent sur le chiffre six.

Dans ce cas particulier, on peut utiliser un test statistique du Chi-2

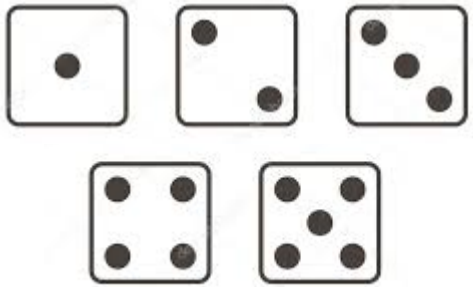
afin de déterminer si j'ai triché.

1/2
1/6

+2*

- Si le dé est équilibré

E(pari) = -1/2 café, je pars perdant



1/2
5/6

-1*

- Si le dé n'est pas équilibré, et tombe la moitié du temps sur le chiffre 6.

E(pari) = (1/2)*(+2 café) + (1/2)*(-1 café)
= 1-1/2 café
= 1/2 café, je pars gagnant

Test statistique du chi-2: Formalisation mathématiques

Le test du chi-deux
d'adéquation à une loi fonctionne
avec n'importe quelle loi de probabilité.
Ici, on se focalise sur le cas de la loi uniforme.

On définit deux hypothèses complémentaires :

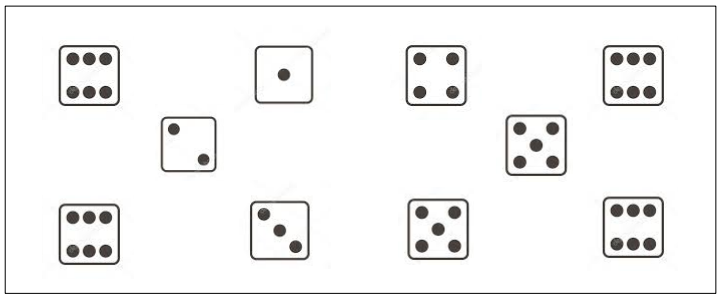
(H0) La distribution des valeurs des dés suit une loi uniforme

→ chaque valeur a la même chance d'être tirée
→ le dé n'est pas pipé

(HA) ou (H1) La distribution des valeurs des dés ne suit pas une loi uniforme

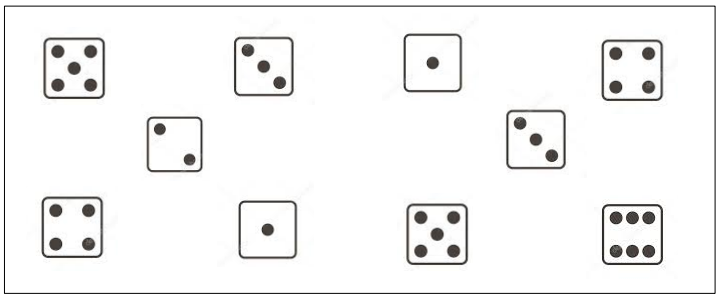
→ Au moins une valeur a une plus grande fréquence d'apparition que les autres
→ le dé est pipé

On va chercher à valider (H0) ou à invalider (H0) au profit de (H1) à l'aide des données sur le tirage des dés.



Plutôt

(H0) ou (H1) ?



Test statistique du chi-2: Formalisation mathématiques (2)

Pour un dé à six faces, on note i (entre 1 et 6 inclus) la valeur du dé.

Pour 30 lancers, on note :

→ O(i) le nombre d'occurences observées de la valeur i

→ E(i) l'effectif théorique que l'on devrait atteindre si (H0) est vérifiée
(dans notre cas de loi uniforme, chaque valeur i vérifie E(i) = 30/6 = 5)

Pratique

i	1	2	3	4	5	6
O(i)	4	7	2	6	5	6
E(i)	5	5	5	5	5	5

Sur les 30 lancers, je comptabilise quatre fois le chiffre 1, sept fois le chiffre 2, etc.

$$D=\frac{(4-5)^2}{5}+\frac{(7-5)^2}{5}+\frac{(2-5)^2}{5}+\frac{(6-5)^2}{5}+\frac{(5-5)^2}{5}+\frac{(6-5)^2}{5}=\frac{1}{5}+\frac{4}{5}+\frac{9}{5}+\frac{1}{5}+\frac{0}{5}+\frac{1}{5}=\frac{16}{5}$$

On trouve dans les tables de loi du chi-deux
q(6-1, 0,05) = 11,07, donc comme D=16/5=3,25 < 11,07=q,
on ne rejette pas (H0), la distribution paraît uniforme.

Dans ce cas-là, je suis sûr de ne pas avoir triché à 95 % !

Théorie

On définit ensuite une statistique de test D, une valeur calculable à partir des données, qui va servir à trancher entre (H0) et (H1).

$$D=\sum_i \frac{(O(i)-E(i))^2}{E(i)}$$

On fixe le risque alpha.
Si alpha est fixé à 5 %, on prendra une décision qui sera valide dans 95 % des cas.
Avec #{i} le nombre de valeurs différentes de i possibles, on définit q le quantile de la loi du chi-deux d'ordre alpha à #{i}-1 degrés de liberté.

D > q → On rejette (H0)

D ≤ q → (H0) n'est pas rejetée

Test statistique : L'essentiel à retenir

- **Deux hypothèses qui s'affrontent ; (H0) et (H1)**
- Dans le cas général, **on se fixe un niveau d'erreur alpha** (souvent 0,05, souvent 0,01 ≤ alpha ≤ 0.1), et on calcule une p-valeur pval

  pval = Probabilité d'obtenir les données observées (ou des valeurs plus extrêmes) en admettant que (H0) soit valide

  pval = P(x|H0)
- **En pratique :**

  → **Si pval ≤ alpha : on rejette (H0)**

  → **Si pval > alpha : on ne rejette pas (H0)**

Test de Student ou t-test simple

- On observe deux groupes p1 et p2 au sein d'une population. Pour une variable quantitative x donnée, on note μ1 la moyenne de x pour le groupe 1 et μ2 la moyenne de x pour le groupe 2. Alors le t-test simple permet de déterminer quelle hypothèse choisir parmi les deux suivantes;

(H0) $\mu_1 = \mu_2$

Les moyennes des deux populations ne sont pas significativement différentes, on peut considérer qu'elles sont du même ordre de grandeur.

(H1) $\mu_1 \neq \mu_2$

Les moyennes des deux populations sont significativement différentes.

P1 P2

Dans l'intuition, (H0) ou (H1) ?

- Pour x le nombre d'infirmières en France
- Pour x le nombre d'infographistes en France

Exemple avec python

Exemple de t-test :

Est-ce que les hommes consomment plus d'alcool que les femmes d'après nos données?

```
alcool_femme = ronfle.query("Sexe=='Femme'")['Alcool']
moyenne_femme = np.mean(alcool_femme)
alcool_homme = ronfle.query("Sexe=='Homme'")['Alcool']
moyenne_homme = np.mean(alcool_homme)
```

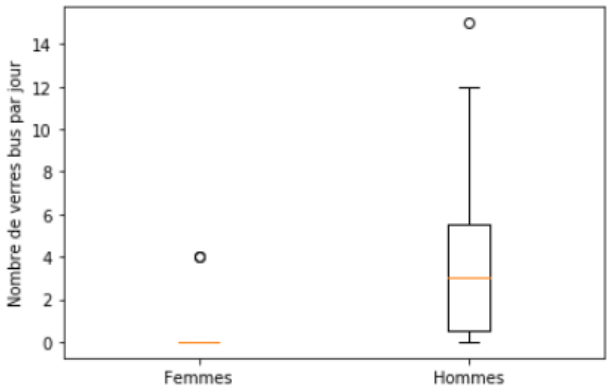
Une visualisation peut aider à y voir plus clair

```
alcool=[alcool_femme, alcool_homme]

fig = plt.figure(figsize=(6, 4))
ax = fig.add_subplot(111)


bp = ax.boxplot(alcool)

ax.set_xticklabels(['Femmes', 'Hommes'])
plt.ylabel('Nombre de verres bus par jour')
plt.show()
```



On reprend le jeu de données sur le ronflement du tp précédent.

D'après ce graphe, (H0) ou (H1) ?

Exemple avec python (2)

Un test pour confirmer l'intuition

```
scipy.stats.ttest_ind(alcool_homme, alcool_femme)
```

```
Ttest_indResult(statistic=5.038109663206171, pvalue=2.1468627505526864e-06)
```

On rejette l'hypothèse nulle; les niveaux d'alcool consommés par les femmes et les hommes sont significativement différents

```
moyenne_homme>moyenne_femme
```

```
True
```

Les hommes consomment significativement plus d'alcool que les femmes sur notre jeu de données

Pour aller plus loin

- Toujours la doc ; https://www.scipy.org/
- Table de loi du chi-deux https://archimede.mat.ulaval.ca/stt1920/STT-1920-Loi-du-khi-deux.pdf
- La statistique expliquée à mon chat, p-valeur ou je fais un malheur! https://www.youtube.com/watch?v=xVIt51ybvu0 → l'une des meilleures chaînes de vulgarisation sur les statistiques
- Culture générale ; https://www.blog-emploi.com/mixite-metiers/