

Informatique 7

numpy, axes, matplotlib

Rémi Emonet - 2020
Université Jean Monnet - Laboratoire Hubert Curien



Informatique 7 : Plan

- Numpy : rappels, agrégations et axes
- Numpy : lecture de csv
- Bibliothèque Matplotlib : premier contact
- Cas d'utilisation
- Matplotlib : principes

3 / 27 - Rémi Emonet - Informatique 7 numpy, axes, matplotlib



IMPORTANT : apprendre par l'expérience

11 / 27 - Rémi Emonet - Informatique 7 numpy, axes, matplotlib



Numpy : fonctions d'agrégations

```
1 a = np.random.uniform(0.95, 1.05, (20, 50))
2
3 print(a.shape) # (20, 50)
4 print(a.size) # 1000
```

python

```
1 print(np.sum(a)) # ≈ 1000
```

python

```
1 print(np.mean(a)) # ≈ 1
2 print(np.var(a)) # ≈ 0.0008333 (0.12 / 12)
3 print(np.std(a)) # = np.var(a) ** 0.5
```

python

```
1 print(np.min(a)) # ≥ 0.95
2 print(np.max(a)) # ≤ 1.05
```

python

- Indice du min/max

```
1 b = np.array([[10, 20, 30],
2              [60, 50, 40]])
3
4 print(np.argmin(b)) # 0
5 bligne = b.reshape((-1,))
6 print(np.argmax(bligne)) # 3
7
8 print(np.argmax(b)) # 3 (comme si en ligne)
9
10 print(np.unravel_index(np.argmax(b), b.shape)) # (1, 0)
```

python

12 / 27 - Rémi Emonet - Informatique 7 numpy, axes, matplotlib



Numpy : axes d'un tableau

- Axes d'un tableau ?
 - un axe est une dimension du tableau (numérotés en partant de 0)
 - *shape* correspond donc aux nombres d'éléments selon chaque axe
- Illustration

```
a = np.arange(21).reshape((3, 7))
```

	Axe 1						
Axe 0	0	1	2	3	4	5	6
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20

```
print(a[2, 0])
print(a[1:, -1])
print(a[1, 2:5])
```

```
1 a = np.arange(21).reshape((3, 7))
2 print(a)
3
4 print("Selon l'axe 0, il y a", a.shape[0], "valeurs")
5 print("Selon l'axe 1, il y a", a.shape[1], "valeurs")
```

python

13 / 27 - Rémi Emonet - Informatique 7 numpy, axes, matplotlib



Numpy : axes et agrégations

- Les fonctions d'agrégation (min, max, sum, mean, ...) acceptent deux paramètres
 - `axis` : l'axe selon lequel agréger (c'est l'axe qui va « disparaître »), avec `np.sum` et un tableau à 2 dimensions :
 - `axis=0` : sommer chaque colonne
 - `axis=1` : sommer chaque ligne
 - `keepdims` : pour décider si l'axe disparaît ou est réduit à un élément
 - `keepdims=False` (par défaut), l'axe disparaît totalement
 - `keepdims=True`, l'axe est conservé, il n'a alors plus qu'un élément

- Exemple

```
1 a = np.ones((3, 7))
2 print(np.sum(a)) # 21
3
4 print(np.sum(a, axis=0).shape) # (7,)
5 print(np.sum(a, axis=1).shape) # (3,)
6
7 print(np.sum(a, axis=0, keepdims=True).shape) # (1, 7)
8 print(np.sum(a, axis=1, keepdims=True).shape) # (3, 1)
9 print(np.sum(a, keepdims=True).shape) # (1, 1)
```

python

- Avancé (pour les tableaux à $n > 2$ dimensions)

- il est possible de passer un tuple à `axis`, par ex, `axis=(0, 1)`

16 / 27 - Rémi Emonet - Informatique 7 numpy, axes, matplotlib



Informatique 7 : Plan

- Numpy : rappels, agrégations et axes
- **Numpy : lecture de csv**
- Bibliothèque Matplotlib : premier contact
- Cas d'utilisation
- Matplotlib : principes



Numpy : lecture de fichier texte

- Fichier de données textuels
 - CSV : *Comma-Separated Values*
 - valeur séparées par des virgules
 - NB: le « . » est utilisé pour les nombres réels en programmation
 - il existe aussi TSV (*Tabulation-Separated Values*)
- Exemple CSV (ici, 3 valeurs par ligne)

```
1 1219,5,89.57
2 1220,9,89.66
3 1221,22,89.88
4 1222,22,90.10
5 1223,8,90.18
6 1224,16,90.34
7 1225,18,90.52
8 1226,13,90.64
9 1227,1,90.65
10 ...
```

- Lecture avec la fonction `genfromtxt` de Numpy

```
1 a = np.genfromtxt('monfichier.csv', delimiter=',')
2
3 print(a.shape) # (... , 3) dans l'exemple
```

python



Informatique 7 : Plan

- Numpy : rappels, agrégations et axes
- Numpy : lecture de csv
- **Bibliothèque Matplotlib : premier contact**
- Cas d'utilisation
- Matplotlib : principes



Matplotlib : exemple

```
1 from matplotlib import pyplot as plt
2
3 x = np.linspace(0, 10, 2000)
4 y = x**3
5 z = (10+x)**2
6
7 plt.plot(x, y, label="courbe 1")
8 plt.plot(x, z, label="courbe 2")
9 plt.legend()
10
11 #plt.savefig('courbes.pdf')
12 plt.show()
13
```

python



Informatique 7 : Plan

- Numpy : rappels, agrégations et axes
- Numpy : lecture de csv
- Bibliothèque Matplotlib : premier contact
- **Cas d'utilisation**
- Matplotlib : principes



Retour sur π : agrégation, courbes



Illustration de Numpy + Matplotlib : affichage des pas (quand on marche)

25 / 27 - Rémi Emonet - Informatique 7 numpy, axes, matplotlib



Informatique 7 : Plan

- Numpy : rappels, agrégations et axes
- Numpy : lecture de csv
- Bibliothèque Matplotlib : premier contact
- Cas d'utilisation
- **Matplotlib : principes**

26 / 27 - Rémi Emonet - Informatique 7 numpy, axes, matplotlib



Pyplot: `from matplotlib import pyplot as plt`

- Matplotlib permet de
 - tracer des figures
 - tracer des courbes, histogrammes, etc
 - sauver dans un fichier image
 - afficher de manière interactive
- `from matplotlib import pyplot as plt`
 - `plt` pour simplifier les tâches simple avec une courbe
- Quelques fonctions
 - `plt.truc(...)`
 - `plot, savefig, show, ...`
 - `scatter, bar, barh, contour, contourf,`
 - `xlabel, ylabel, xticks, yticks, legend, grid, title, ...`
 - `text, annotate, ...`
 - `imshow, ...`
- http://matplotlib.org/api/pyplot_summary.html

27 / 27 - Rémi Emonet - Informatique 7 numpy, axes, matplotlib

