Matplotlib

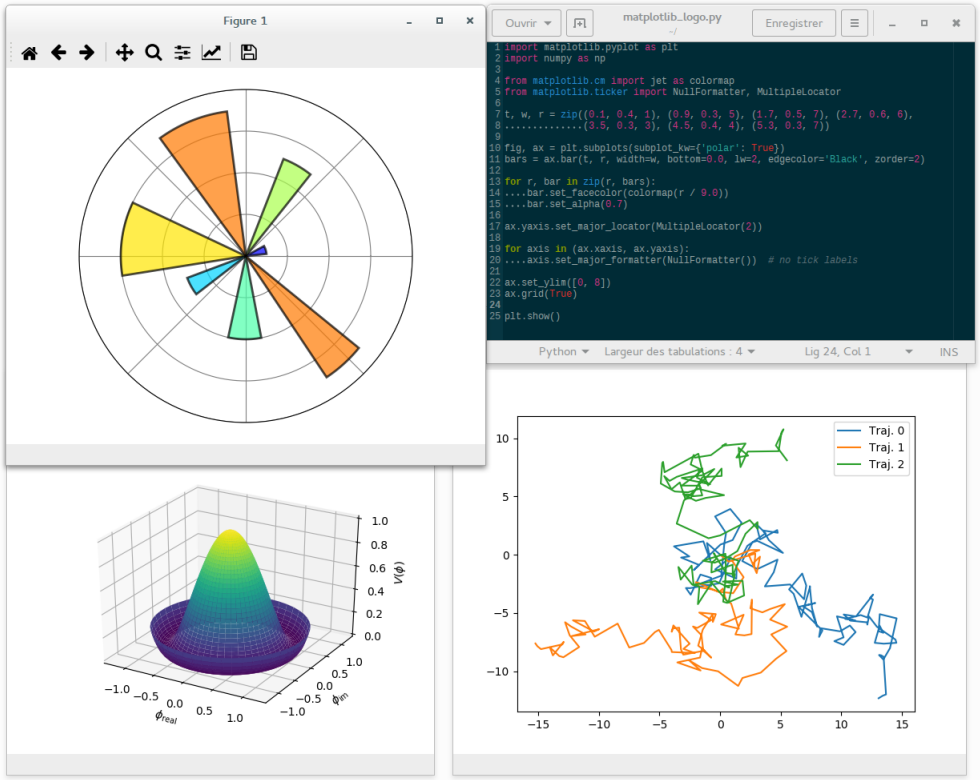
Cours M2 CCN

Modélisation des problèmes scientifiques

2019/2020

À quoi ça sert ?

Librairie python dédiée à l'affichage graphique (courbes, boxplot, 3d, etc.)



→ Sur le plan professionnel ; quand on parle à des clients, pour présenter des résultats plus lisibles que des données brutes ou des gros tableaux

→ Benchmarks

→ 1 graphe > 10 explications

Étude du ronflement

Matplotlib par l'exemple :

- On va étudier différents graphiques relatifs à un jeu de données fourni par le CHU d'Angers sur le ronflement

  (voir ici ; http://www.info.univ-angers.fr/~gh/Datasets/ronfle.htm)

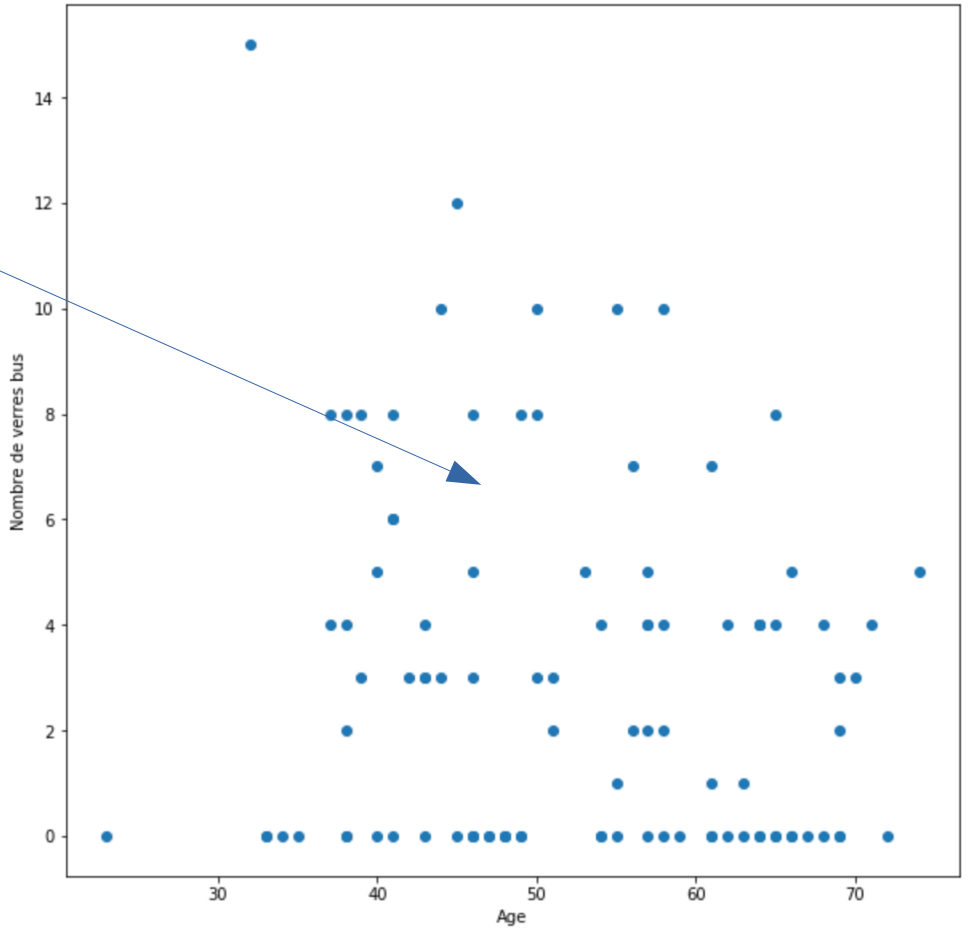
	Age	Poids	Taille	Alcool	Sexe	Ronflement	Tabac
P0001	47	71	158	0	Homme	Non-Ronfleur	Fumeur
P0002	56	58	164	7	Homme	Ronfleur	Non-Fumeur
P0003	46	116	208	3	Homme	Non-Ronfleur	Fumeur
P0005	70	96	186	3	Homme	Non-Ronfleur	Fumeur
P0006	51	91	195	2	Homme	Ronfleur	Fumeur

Variable quali/quanti ?

Nuage de points

```
import matplotlib.pyplot as plt
```

```
plt.scatter(ronfle['Age'], ronfle['Alcool'])
plt.xlabel('Age')
plt.ylabel('Nombre de verres bus')
plt.show()
```



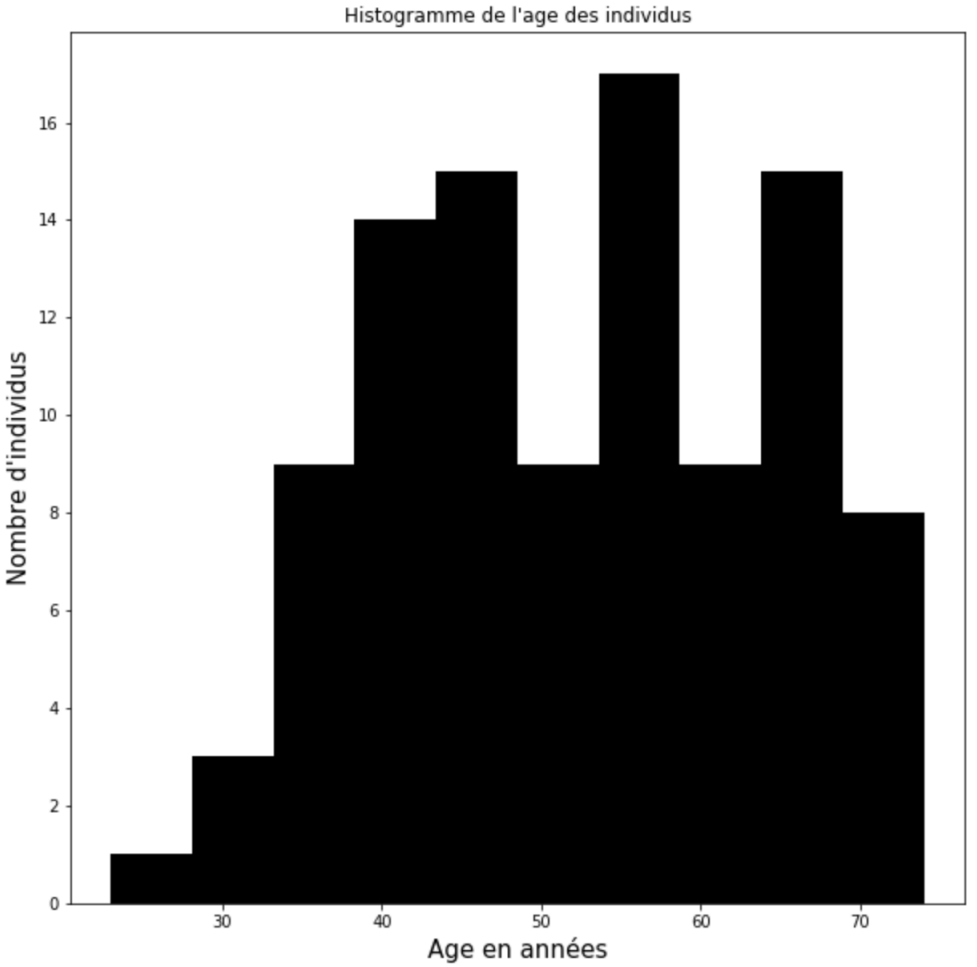
On importe matplotlib.pyplot, qui gère les affichages de graphes standards.

plt.scatter correspond au nuage de points
plt.xlabel correspond à la légende de l'axe x
plt.ylabel correspond à la légende de l'axe y
plt.show() affiche le graphique défini par les commandes précédentes

Quelques commandes utiles sur jupyter :
%matplotlib inline pour afficher les graphes sans plt.show
%matplotlib notebook pour afficher les graphiques interactifs

Histogramme

```
plt.figure(figsize = (10,10))
plt.hist(ronfle['Age'], color = 'black')
plt.xlabel('Age en années', size = 15)
plt.ylabel("Nombre d'individus", size = 15)
plt.title("Histogramme de l'age des individus")
plt.show()
```



plt.figure permet de spécifier certaines options, comme la taille de la figure (figsize).

Pour construire un histogramme → plt.hist

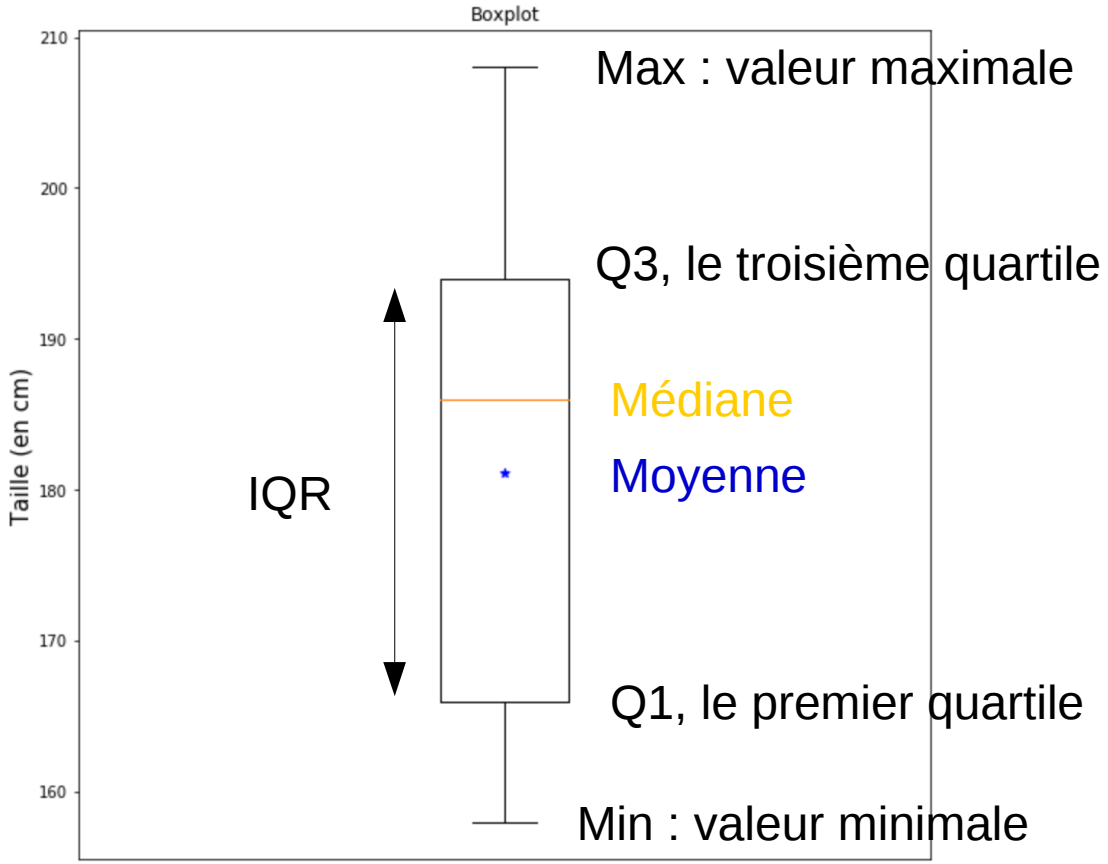
On peut ajouter un titre à la figure à l'aide de plt.title

La majorité des fonctions de matplotlib ont un paramètre color, qui gère le choix de la couleur (voir http://www.python-simple.com/img/img41.png pour les choix de couleurs)

Les tailles des polices de légende peuvent être paramètrées à l'aide de l'option size.

Diagramme de Tukey (Boxplot)

```
plt.figure(figsize=(10,10))
plt.boxplot(ronfle['Taille'])
plt.scatter(1,np.mean(ronfle['Taille']), marker ="*", color="blue") # Ajout de la moyenne (l'étoile)
plt.title("Boxplot")
plt.xticks([]) # Pas de graduation des valeurs pour l'axe des abcisses
plt.ylabel("Taille (en cm)", size=15)
plt.show()
```



Dans le cas de valeurs aberrantes (trop grandes/petites par rapport au reste des donnnées), le maximum correspond à la valeur non aberrante la plus grande du jeu de données.

InterQuartile Range = Q3 - Q1

Seuil min = Q1-1,5*IQR
Seuil max = Q3+1,5*IQR

Les valeurs aberrantes sont alors représentées sous forme de points.

Diagramme circulaire

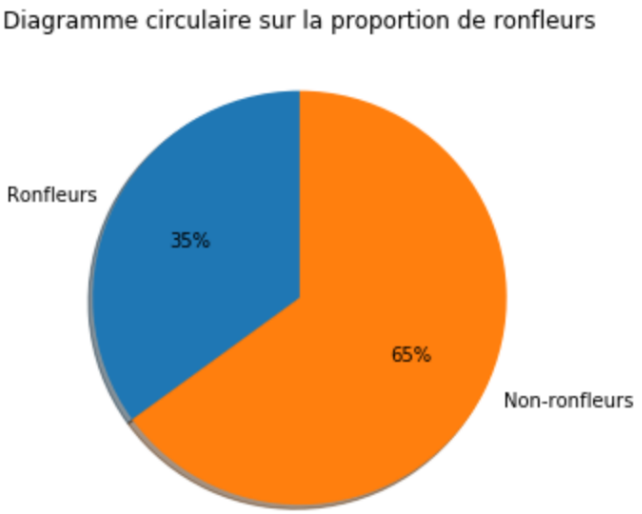
```
ronflement = ronfle['Ronflement']

nbNonRonfleur = len(ronflement[ronflement == "Non-Ronfleur"])
nbRonfleur = len(ronflement[ronflement == "Ronfleur"])

compteRonfleurs = np.array((nbRonfleur, nbNonRonfleur), np.int64)

#On crée un tableau comptant le nombre d'occurences de chaque modalité

plt.figure(figsize=(5,5))
plt.pie(compteRonfleurs,
        startangle = 90,
        # Par convention, on commence par la proportion la plus importante à un angle de 90°
        # Puis on classe les différentes proportions par ordre décroissant
        labels = list(("Ronfleurs", "Non-ronfleurs")),
        autopct='%1.f%%', # Arrondi du pourcentage à l'entier
        shadow = True)
plt.title("Diagramme circulaire sur la proportion de ronfleurs")
plt.show()
```



Parfois, certains graphes simples d'apparence peuvent nécessiter de modifier les données. L'esthétique du graphe peut prendre quelques lignes, ne pas hésiter à aller regarder les options des différentes fonctions dans la doc !

pandas.DataFrame.plot(pie) marche aussi très bien !

Pour aller plus loin

- D’autres types de graphiques plt.lines
- Par exemple, voir ce tutoriel : https://www.codingame.com/playgrounds/17176/recueil-dexercices-pour-apprendre-python-au-lycee/cours---representation-graphique-avec-matplotlib
- Voir la gallerie pour s’inpirer ; https://matplotlib.org/3.1.0/gallery/index.html
- Définir des sous-graphes avec plt.subplot
- Maîtriser les éléments de son graphique (axes, couleur, transparence, légende, etc.)
- En bonus : afficher des images (plt.imshow), graphiques en 3d, cartographie
- Toujours en bonus : Représenter des fonctions et des formes géométriques

  Voir http://desaintar.free.fr/python/tp/tp_graphiques.pdf
- Encore en bonus : Graphes interactifs avec plt.ion