

Enseignant :

- Joseph Salmon : joseph.salmon@umontpellier.fr,

Horaire de consultation (*office hours*) :

- Jeudi 18h-19h, sur rendez-vous par mail uniquement.

Page web du cours :

Supports : <http://josephsalmon.eu/HMMA238.html>

Prérequis

Les étudiants sont supposés connaître les bases de la théorie des probabilités, de l'optimisation, de l'algèbre linéaire et des statistiques.

Description du cours

Ce cours traite de l'utilisation de Python, avec pour objectif de pouvoir effectuer du traitement et de la visualisation de données avec ce langage. Nous nous intéresserons principalement aux méthodes basiques de programmation en Python ainsi qu'aux bibliothèques standards de méthodes numériques (`numpy`, `scipy`) de traitement de bases de données (`pandas`) et de visualisation (`matplotlib`, `seaborn`). Au delà des aspects de programmation, on veillera aussi à introduire de bonnes pratiques numériques (tests, gestionnaire de versions, etc.).

Notation : Projet = 1/3 de la note finale de HMMA238

- À effectuer par groupes de 2 ou 3 uniquement.
- Date de mise en ligne du sujet : 28 avril.
- Date de rendu (à envoyer par mail) : Dimanche 26 Mai 23h59 (attention : 0 en cas de dépassement).
- **Rendu attendu** : Un lien `github` pointant vers votre travail, et qui soit "clonable". Le dépôt `github` devra contenir un fichier `readme` dans l'accueil. Le dépôt devra montrer une utilisation équilibrée entre membres de chaque groupe (les notes pourront être différenciées par élève si le besoin s'en fait sentir).

Éléments nécessaires pris en compte dans la notation :

- la création d'un dossier "code" qui contiendra un fichier `utils.py` avec les fonctions utilisés, un notebook de synthèse. Le notebook devra être exécutable, et donc permettre de reproduire les résultats décrits (*i.e.*, *Restart & Run all* marche correctement).
- la création d'un dossier "latex" : dans ce dossier sera versionné un fichier `rapport.tex` qui décrira votre démarche dans la résolution du problème, ainsi que les fichiers nécessaires à la compilation du `.pdf`¹ (par exemple les fichiers de bibliographies, d'images, etc.). Sur ce point de l'aide se trouve ici :
<http://josephsalmon.eu/enseignement/Montpellier/HMMA238/draft-project.zip>
- la pertinence des réponses aux questions

1. Un conseil : ne versionnez par le fichier `rapport.pdf`...

- la qualité des graphiques (légendes, couleurs)
- l'utilisation du style PEP8
- la lisibilité du code produit (noms de variable clairs, commentaires utiles, code synthétique, etc.)
- un protocole de test des algorithmes considérés
- l'absence de bug
- la qualité de rédaction et d'orthographe

Plus de détails seront donnés dans l'énoncé du projet.

Thèmes des séances (si le temps le permet)

- Introduction : problématique du développement logiciel, logiciels jupyter notebook / jupyter lab. Environnements de développement intégré
- la programmation : algorithmes, modules, fonctions, objets, variables, vecteurs, tableaux, listes, dictionnaires, conditions, boucles
- efficacité en temps, en mémoire, compilation "just in time" (numba)
- Principes et pratique du débogage : ipdb
- git et github et sur pytest pour les tests, travis comme exemple
- Documentation : sphinx / doxygen

Bonus

2 pt supplémentaires sur la note de la partie Python peuvent être obtenus pour toute contribution à l'amélioration des cours (notebooks, latex, etc.) sous les contraintes suivantes.

Contraintes :

- on obtient 1 point par amélioration
- seule la première amélioration reçue est "rémunérée", les autres ne gagnent plus rien
- il faut déposer un fichier **.txt** (taille <10 ko) en créant une fiche dans la partie du Moodle intitulée "Bonus - Proposition d'amélioration"
- détailler précisément (ligne de code, page des présentations, etc.) l'amélioration proposée, ce qu'elle corrige et/ou améliore
- pour les fautes d'orthographe : proposer au minimum 5 corrections par contribution
- chaque élève ne peut gagner que 2 points maximum avec les bonus

Livres et Ressources en ligne complémentaires

Général : Skiena, The algorithm design manual, 1998 (en anglais) [[Ski98](#)]

Général : Courant *et al.*, Informatique pour tous en classes préparatoires aux grandes écoles : Manuel d'algorithmique et programmation structurée avec Python, 2013, [[CdG+13](#)]

Général / Science des données : Guttag, Introduction to Computation and Programming, 2016 [[Gut16](#)] (en anglais)

Science des données : J. Van DerPlas, *With Application to Understanding Data* [Van16] (en anglais)

Code et style : Boswell et Foucher, *The Art of Readable Code* [BF11] (en anglais)

Python : <http://www.scipy-lectures.org/>

Visualisation : exemple de visualisation <http://openclimatedata.net/>

Références

- [BF11] D. Boswell and T. Foucher. *The Art of Readable Code*. O'Reilly Media, 2011.
- [CdG⁺13] J. Courant, M. de Falco, S. Gonnord, J.-C. Filiâtre, S. Conchon, G. Dowek, and B. Wack. *Informatique pour tous en classes préparatoires aux grandes écoles : Manuel d'algorithmique et programmation structurée avec Python*. Eyrolles, 2013.
- [Gut16] J. V. Guttag. *Introduction to Computation and Programming Using Python : With Application to Understanding Data*. MIT Press, 2016.
- [Ski98] S. S. Skiena. *The algorithm design manual*, volume 1. Springer Science & Business Media, 1998.
- [Van16] J. VanderPlas. *Python Data Science Handbook*. O'Reilly Media, 2016.